The Dissertation Committee for Michael James Van Wie Certifies that this is the approved version of the following dissertation:

DESIGNING PRODUCT ARCHITECTURE: A SYSTEMATIC METHOD

Committee:

Kristin L. Wood, Supervisor

Matthew I. Campbell, Co-Supervisor

Billy V. Koen

Richard H. Crawford

Arthur B. Markman

Robert B. Stone

DESIGNING PRODUCT ARCHITECTURE:
A SYSTEMATIC METHOD

by

Michael James Van Wie, BS, MSE

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December, 2002

Dedication

To my parents and my wife Kimi.

DESIGNING PRODUCT ARCHITECTURE: A SYSTEMATIC METHOD

Publication No.____________

Michael James Van Wie, PhD

The University of Texas at Austin, 2002

Supervisors: Kristin L. Wood and Matthew I. Campbell

Architecture design, also thought of more loosely as layout design within the context of conceptual design, is one stage of the mechanical design process that significantly impacts product performance in terms of manufacturing, assembly, modularity, product family variety, maintenance, etc. This design step is special because it marks an occasion when many effects, including geometric concerns, come into play simultaneously on a large scale. The purpose of this research is to investigate the architecture design phase and develop a new design method as there is currently no consensus regarding a best strategy for dealing with architecture design. The resulting method is based primarily on the development of a formal representation and a set of guidelines derived from an empirical product study. Each of these three main deliverables are assessed and validated as part of their development.

Based on the concept of a mental model, a representation is developed which consists of a lexicon and a six element notation known as the architecture workframe. Terms of the lexicon provide a well-defined means to describe

various aspects of architecture while the notation instantiates these terms in a reasonable format in order to facilitate effective manipulation of the architecture. This representation allows the designer to incrementally proceed from initial constraints to a fully described layout at the conceptual level. The representation directly supports design for modularity and design for flexibility. Effectiveness of the representation is confirmed through an experimental comparison of this technique with an analogous conventional method. Results are promising in terms of the quantity, quality, and efficiency of design solution. In working toward the second deliverable, an empirical study of thirty product evolutions is performed and ten guidelines are extracted through a process of making observations, hypothesizing guidelines, and refining a set of guidelines. Validation of these guidelines is performed using a second sample set of existing products that are representative of the larger population of products. Finally, a cohesive method is proposed to encapsulate the representation and guidelines into a design strategy. The method is assessed with respect to method constituents and the expected bounds of performance of those constituents.

Table of Contents

List of Tables

List of Figures

Chapter 1 – Introduction to Product Architecture

Product design is the act of synthesizing a physical solution in order to satisfy a set of customer needs. This task is a messy process. Several difficulties arise especially during the initial development of concepts and the initial development of device layout. The fundamental problem with this particular task is that the designer is faced with an overwhelming number and range of issues to consider within a relatively compressed stage of the overall design process. A consequence of this problem is a general deficit in the designer's ability to address relevant design issues and develop satisfactory architecture solutions in a timely manner. The purpose of this research is to examine the architecture design stage and develop a systematic method for improving the designer's ability to more directly achieve design solutions and more efficiently respond to a large set of design issues that characterize architecture design.

1.1 ARCHITECTURE DESIGN - PROBLEM DESCRIPTION

The motivation for this work is rooted in the conventional practices that make architecture design difficult. This section illustrates the challenges of architecture design and shows why an improved design method is an answer to those challenges.

1.1.1 The Architecture Design Process

Architecture design occurs in the context of other design tasks. A design process model provides a view of typical activities in the overall act of design. An abbreviated version is a three-step sequence: requirements, function, and form. Many models, some more detailed than a three-step process, are found in the literature (Pahl and Beitz, 1996; Ulrich and Eppinger, 2000; Roozenburg and Eekels, 1995; Otto and Wood, 2001; IEEE-1220, 1998). Based on these prior models, Figure 1.1 describes a generic design process. Design is inherently iterative and therefore feedback loops are not shown explicitly in the model although they certainly exist both among and within steps of the design process. Table 1.1 defines terms related to the figure.

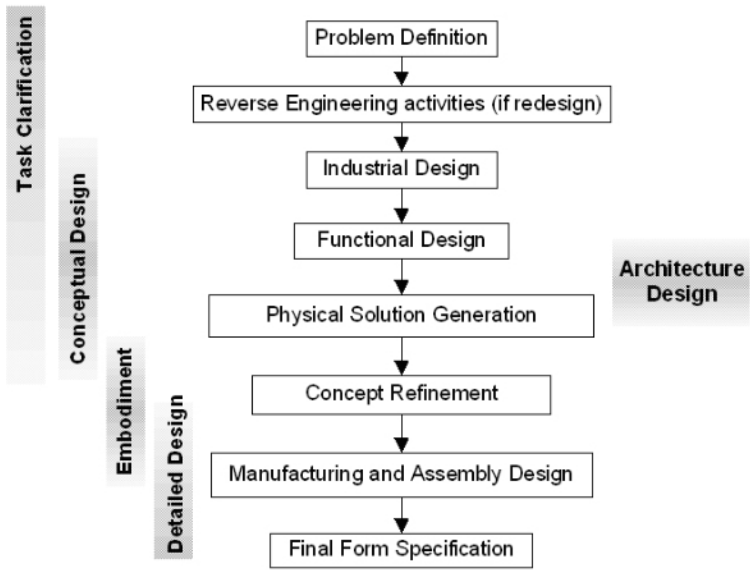


Figure 1.1 Design Process Model

Table 1.1 Design Activities

Term	**Description**
Architecture Design	Transformation of function to form; Design of preliminary physical form solutions based on the functional design; typically an implicit activity of conceptual design and embodiment.
Functional Design	Design of ***what*** a design solution does, not ***how***; a design solution independent of physical form.
Conceptual Design	Design of solutions typically at a high level of abstraction; occurs in terms of both function and physical specifications; usually a divergent, creative, and generative process.
Embodiment	Design of physical solutions; typically a refinement and development activity; usually involves a ramping up of modeling and analysis activities.
Detailed Design	Design and specification of all required design variables; usually involves a large degree of modeling and analysis.

The architecture design phase in Figure 1.1 is shown as a shaded transition that begins during functional design, is prominent through physical solution generation, and ends before embodiment design. Figure 1.2 highlights this phase and shows the typical inputs and outputs of architecture design. The architecture design process may be viewed as a layout task where the designer must create a spatial solution that is eventually defined as an arrangement of components and assemblies.

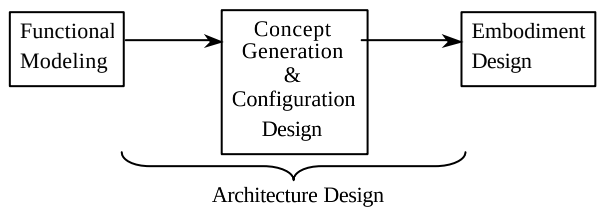


Figure 1.2 Architecture Design

Between functional design and embodiment, architecture emerges informally during concept generation phase and becomes an explicit concern during configuration or layout design (Ulrich and Eppinger, 1995). Based on the descriptions in Table 1.1, one can see that modeling and analysis activities are characteristic of detailed design and, to a somewhat lesser extent, embodiment design.

When one considers conceptual design, however, there is a glaring lack of modeling and analysis generally used in practice. For example, engineers have at their disposal techniques to rigorously address the detailed design question of "How big should this shaft be to support some load?" By using an appropriate solution from the vast collection in the engineer's analytical toolbox, this problem is manageable. On the other hand, if the designer is faced with the problem of developing an initial layout that includes major modules and components, the number of relevant and appropriate tools in the designer's toolbox now seem to dwindle to a much smaller set. The designer may simply apply ad hoc techniques like sketching out a few alternatives that appear to be good according to his own tacit knowledge that is based on his prior experience. The prototypical engineer is famous for solving problems in a logical and rigorous style. Unfortunately, the engineer's modeling and analysis capabilities are not generally used in architecture design. This is one principle motivation for developing a design method for architecture design.

One recurring problem in architecture design is the difficulty in dealing with both function and form. Given the spatial effects associated with form solutions, a large number of design issues arise during this stage as shown in Table 1.2.

Table 1.2 Architecture design factors

Part number and complexity
Manufacturing and assembly
Product family variety
Standardization
Modular vs. integral
Interfaces
Serviceability / maintenance
Industrial design

The task of architecture design becomes very complex as the designer attempts to keep track of these items. Without clear direction, the search for solutions is a cumbersome and inefficient path. An analogy of this complication is illustrated in Figure 1.3.

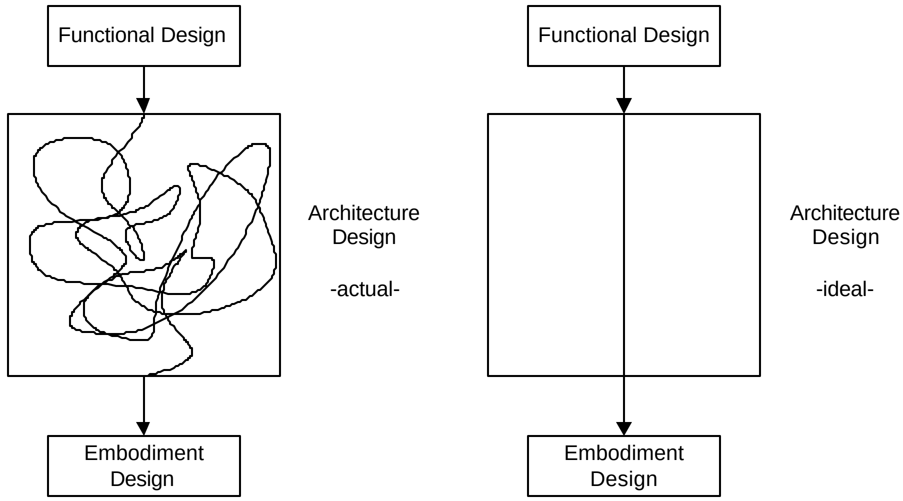


Figure 1.3 Model of the architecture design process

It may seem odd that the actual design path modeled above deviates greatly from the ideal case. After all, the purpose of architecture design is very clear. The designer must simply find a preliminary physical solution (generally multiple alternatives) that satisfies some functionality. A closer look at products themselves should clarify the problem of this task.

1.1.2 Artifacts of Architecture Design

The discussion, to this point, refers mainly to the ***process*** of architecture design with little mention of the substance of architecture itself. What exactly does product architecture look like? What is the result of performing architecture design? There are several valid answers depending on one's viewpoint. Consider a tangible perspective–visual inspection of a device as dissected in Figure 1.4. Two similar cordless screwdrivers with different architectures are shown.

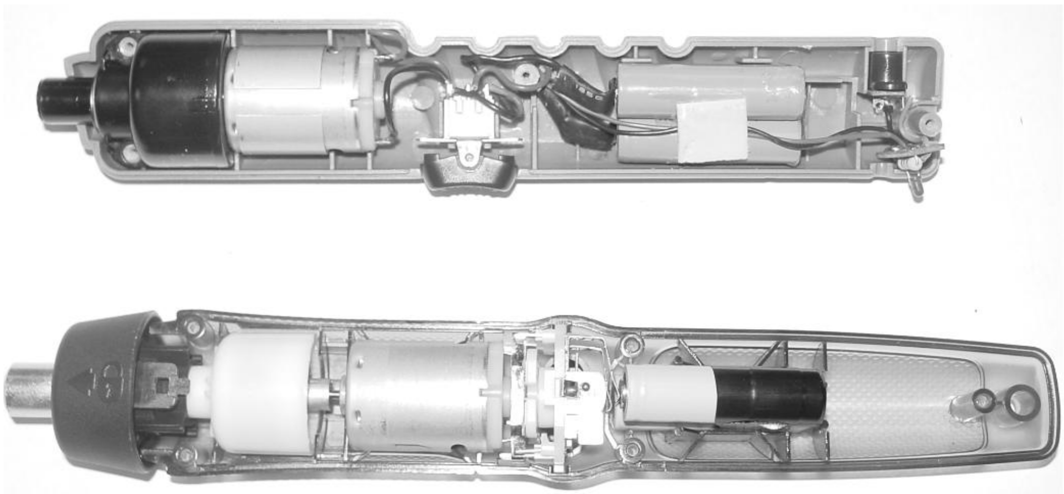


Figure 1.4 Example of two different architectures

The device in the photograph includes a great amount of detailed design information in addition to the architecture. If this extraneous information is removed, what is left in terms of architecture? ***Fundamentally, architecture is about a set of items and how they are arranged.*** But what are the items, and how is their arrangement described? One can argue that the physical modules such as the chuck, gearbox, motor, switch, batteries, and housing form a set of elements, and the spatial layout as shown defines their arrangement. One could even create an abstraction such as a graph to represent this description of architecture. This approach is one of many valid perspectives, representing a significant challenge of architecture design today. There is no widely accepted formal vocabulary and grammar for describing product architecture.

Previous research has left us with several versions of architecture constructs including modules, interfaces, working principles, wirk elements (physical aspects corresponding to some functionality), etc. (Stone, 1997; Sosa et al., 2000; Pahl and Beitz, 1996; Jensen, 2000). Most of these concepts have some advantages usually in representing some specific aspect of architecture design such as function sharing. However, in the interest of describing product architecture with respect to all its principle elements, as partially listed in Table 1.2, a basic problem still remains: product architecture is an ill-defined design concept, and there is a need for a representation that captures those items which are important in helping the designer design.

This lack of an architecture language is one reason for the roundabout design path shown in Figure 1.3. Specifically, the problem is that the designer must consider a large set of issues without a vehicle for incrementally describing and thus keeping track of the design situation. This scenario is analogous to one attempting to develop a Broadway show without having such a thing as a casting list, action script, or map of the stage. It is simply difficult to manage the leap from function to form without a language for describing and thus keeping track of the design situation.

In addition to the problem of representing the artifacts of product architecture, how does the designer know what to do with them? Considering again the two screwdrivers in Figure 1.4, which is a better architecture? The answer depends on a multitude of factors including customer needs, manufacturer needs, business constraints, etc. Despite this dependency, perhaps there are "rules of thumb" (heuristics) regarding architecture that generally apply to most products. Given a current lack of design knowledge that addresses this particular concern, there is a need for a set of knowledge that helps guide the designer toward a good solution.

By now it should be more clear as to why architecture design can be described as a circuitous and drawn out process as indicated in Figure 1.3. The two main problems are a lack of a design language and a lack of design knowledge for what makes a good product architecture.

1.1.3 Current Solutions

Despite the grim circumstances that appear to exist during architecture design, the situation is not beyond salvage. As the following discussion shows, the design research community has addressed the architecture design problem in varying degrees, as explored further in Chapter 2. For now, a small cross-section of candidate design processes is presented in the following paragraphs in order to briefly show how the problem has been addressed in the past.

The Pahl and Beitz (1996) approach to product design is to progress from a function structure to working principles, to working structures, then concept variants, and finally reach embodiment design. Figure 1.5 shows the progression.

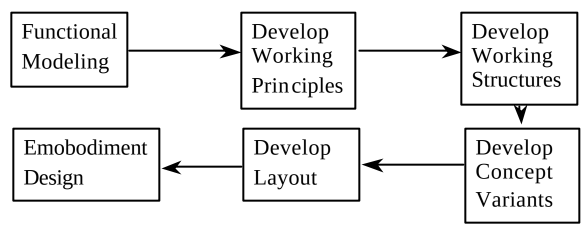


Figure 1.5 Pahl & Beitz (1996) approach

This particular approach is rooted in the notion that a product can be modeled with three primary structures: a function structure, working structure, and a physical structure. In this methodology, layout is performed after the concept variants are developed and certainly after several form issues have been introduced.

Ulrich and Eppinger (2000) as well as Cutherell (1996) propose a different procedure that focuses more on the concept of "chunking" or "clustering" elements of the design. Table 1.3 shows the steps involved in this method.

Table 1.3 Ulrich and Eppinger (2000) approach

1. Create schematic containing elements of both function and form.
2. Cluster elements of the schematic
3. Create a rough geometric layout
4. Identify fundamental and incidental interactions

In this approach, the schematic is a type of combined graph with nodes that represent either functions or physical chunks. In this case, the physical layout of function and form elements take place simultaneously.

Otto and Wood (2001) use a similar methodology as the previous approach, and additionally include a more directed search for chunks or modules during the clustering phase of functional modeling. Broadly, their plan includes three steps as given in Table 1.4. A notable extension to previous methods is the use of three modular heuristics in order to determine modules in the functional model (Stone, 1997). They also prescribe a sequence of concept generation activities that includes rough layout and specific part consideration.

Table 1.4 Method from Otto and Wood (2001)

1. Develop a functional model
2. Apply modular heuristics to the functional model
3. Generate concepts for the modules a. Create rough geometric layouts b. Search for existing components c. Search for creative modules d. Reflect

These approaches illustrate a sample of prior work in which each technique addresses the architecture design problem in a different manner. Further background information is given in the next chapter. All of the above methods have in common a lack of continuity during the leap from function to form. Through the development of techniques such as chunking and modularity, progress has been made toward establishing a reasonable decomposition of the design problem. However, there still remains a need for advanced methods that can facilitate a more smooth transition from function to form. The philosophy of this thesis is to zoom in on this transition step and generate a method for taking incremental small steps instead of a one giant leap.

1.1.4 Goal and Vision

The motivation for this research is based on three main premises. The first is that architecture design is an important aspect of design, whether original or redesign, deserving of techniques that facilitate effective and efficient search and development of design solutions. Recent prior work supports the importance of architecture design (Cutherell, 1996; Otto and Wood, 2000; Stone, et al., 1998; Ulrich and Eppinger, 2000). The second premise is that current architecture design techniques do not measure up to the standard of effective and efficient execution compared with other modeling and analysis techniques for other phases of the design process, such as functional design, Design for Manufacturing and Assembly, Robust design, etc. The third premise is that a systematic method for architecture design is an effective solution to help steer the designer toward solutions. Erkens is credited with pioneering the systematic approach for design in the 1920's (Pahl and Beitz, 1996). Since then, an increasing number of methods have been developed to support the design process. Given the widespread and continuing use of various methods in education and industry, it is clear that a systematic method can be a very useful aid for handling design problems.

The goal of this research is to develop a method for architecture design that guides the designer along a more direct path from function to form as illustrated in Figure 1.6. The fundamental improvement sought is a reduction in design effort required to establish product architecture design solutions.

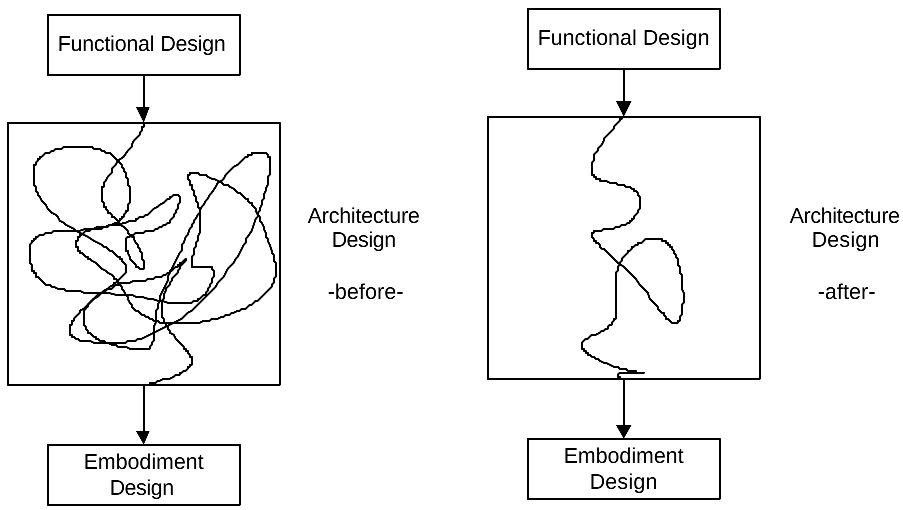


Figure 1.6 Architecture Design path without and with a systematic design method.

In order to meet this level of performance, this work proposes a method to empower the designer through three key constructs: 1) a representation for product architecture, 2) architecture design guidelines, and 3) a method which is a sequence of steps to apply the representation and guidelines. The representation is the method backbone through which the design solutions are manipulated by abstraction, documentation, observation, and control. The representation filters out extraneous information and presents the designer with a somewhat canonical form of product architecture. While inherently useful in giving the designer a workframe in which to develop concepts, the representation lacks explicit guidance regarding the form of the solutions under development. For this reason, a set of guidelines is developed that directs the designer toward a good architecture design. These guidelines show how to manipulate the information offered by the architecture representation. Both the representation and guidelines are presented in a cohesive strategy or method. This method utilizes the representation and guidelines in a sequence of steps and is referred to as the architecture design method.

1.2 HYPOTHESIS AND OBJECTIVES

The hypothesis is that a method based on 1) a formal representation of product architecture and 2) a set of guidelines can lead the designer to architecture design solutions more efficiently in terms of both quantity and quality than conventional design practices. A formal representation facilitates this effect by providing the designer with an inventory of observable and controllable parameters through a design lexicon. This lexicon, realized by a six element notation, allows the designer to focus on those factors that are important while avoiding overly extraneous information that normally imposes a burden during architecture design. In addition to a more direct path to design solutions, the representation provides a basis for identifying physical modules and for measuring product flexibility. In both cases, this knowledge is formed into design guidelines. In addition to these guidelines that aid in designing for modularity and flexibility, other design knowledge based on a product evolution study is codified into guidelines. The following objectives summarize these areas of work:

1. to develop a formal representation for product architecture,
2. to perform an empirical study of devices to form the basis for design guidelines,
3. to develop architecture design guidelines from the empirical study,
4. to develop a method for architecture design, and
5. to perform an original design using the proposed architecture method.

1.3 ORGANIZATION

The layout of this document is partitioned along the lines of the main objectives. The main deliverables include three chapters that present the representation, guidelines, and the method.

Chapter 2 provides the main background shared among the remaining chapters. This chapter covers related prior work and development of the concepts that form the basis for Chapters 3, 4, and 5.

Chapter 3 presents the architecture representation including the procedures for generating a representation. An example is given to illustrate representation capability.

Chapter 4 presents both the empirical study and the guidelines. The empirical study is shown as a direct precursor to guideline development and the full set of guidelines are enumerated in the chapter. Validation of the guidelines is given with respect to a set of existing products.

Chapter 5 gives the process in which the representation and guidelines are used within the context of the overall design process. Method validation is presented including a comparison of the technique with a conventional method. Additionally, the results of an original design using the method are given.

Conclusions and future work are given in Chapter 6. Appendices and a Bibliography follow this chapter.

Chapter 2 – Background

This chapter provides a source of prior work used as the foundation for the concepts presented in the next three chapters. The discussion begins with a research approach for developing design methods. Subsequent sections treat the background of design methods, guidelines, and representations. The objective of this chapter is to give a more concrete foundation for this research than was given in the problem overview from Chapter 1.

2.1 RESEARCH APPROACH

In order to meet the objectives specified in Chapter 1, an approach is adopted from prior work that describes a process for developing design methods. This process was developed by Wood and Greer (2002), and is based on work from Blessing et al. (1998). Only a portion of their model is used as illustrated in Figure 2.1. This model serves as the approach for development of the representation, guidelines, and the method. Although the process is not complex, it defines the scope of actions required to meet the objectives.

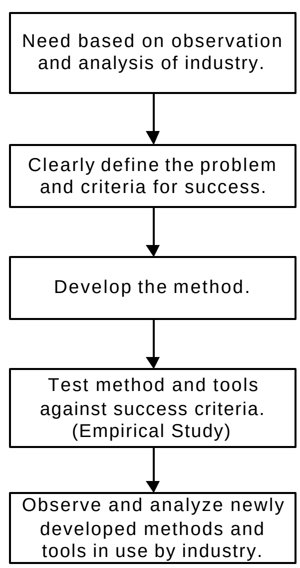


Figure 2.1 Research approach

2.2 DESIGN METHODS

In the early days of design method development, design was viewed more as a craft than a systematic process and few constructs were available for representing abstract ideas of design tasks and design artifacts (Pahl and Beitz, 1996). During the last half of the twentieth century, there was a steady increase in contributions toward the modeling, understanding, and development of design methods. These methods vary widely in their assumptions, theoretical basis, analytical techniques, and scope of application.

Despite this variation, commonalities and persistent themes are evident among these prior efforts. The notion of design domains such as function and form is a consistent perspective maintained throughout previous work. Similarly, concept generation is a design task frequently addressed in the literature. While the term 'concept generation' is somewhat ambiguous as to the stage of the design process involved (ie. function, form, or both?), architecture design is well defined as the transformation from function to form (Otto and Wood, 2001). A few researchers have developed methods for architecture design as those discussed in the previous chapter. These methods qualify as advanced methods relative to the conventional practice since they do offer some degree of problem decomposition. This work seeks to extend the progress from these efforts through the development of an improved architecture design method.

2.3 REPRESENTATIONS

Historically, representations have played an important role in design. Early work by Girard Desargues in the seventeenth century noted the spatial representations that architects and engineers used to improve traditional ad hoc craft techniques (Antonsson and Cagan, 2001). Later in the nineteenth century, Durand developed abstract representations that implemented a spatial framework of grids and axes upon which construction elements could be located and combined in a hierarchical manner (Mitchell, 2001). This approach allows a systematic process to architecting. Machine design benefited similarly when Franz Reuleaux established kinematic elements and combinatory notations (Antonsson and Cagan, 2001). These three representation efforts capture the spirit of representing a complex problem with a simplified model.

In the recent past, techniques of varying levels of formalism have been developed to represent different levels of a product design solution. Two of the many representations will be highlighted: i) function based, and ii) form based. Function based representations allow designers to enjoy the benefits of form independent representations of design solutions. Formal definitions for function structure elements and the function structure generation process have been developed and continue to be refined (Hirtz et al., 2002). On the other end of the spectrum, CAD conceptual modelers (Thompson, 2001) and shape grammar work have focused on representing physical form in terms of its basic property–shape (Stiny, 2001).

A representation for architecture design serves to help a designer be aware of relevant issues, but not extraneous information. This concept is in line with Einstien's remark that a model should be made as simple as possible but not simpler. The following presents a series of different perspectives taken by researchers in engineering design to illustrate the general understanding of and approach to architecture design in recent years.

Welch and Dixon (1992) developed behavior graphs that are based on research in qualitative physics. A solid contribution of this work is a representation that explicitly defines the connectedness of physical embodiments using functional parameters. For purposes of conceptual design, this behavior graph approach is very capable in terms of defining the physical topology of a conceptual solution. Similarly, Aguirre-Esponda (1992) developed a configuration model that mapped a small set of 'ideal' functional elements to a set of physical device elements. This configuration model supported design activities from concept generation to embodiment. One drawback to both approaches is the lack of substantial spatial information regarding the partitioning of spatial regions into components and modules. Despite this limitation, Campbell (2000) extends upon Welch and Dixon's (1992) work to further develop two computational structures known as a functional parameter and an embodiment. This framework strongly supports automated design.

Rosen (1996) takes a different approach by developing a combinatoric method to address configuration design in terms of three discrete design spaces including material compatibility, connections (physical contacts between components), and covers (the relationship of covering a component with others). Rosen's framework is intriguing because he applies the tools of discrete mathematics to the architecture problem and therefore brings to bear a rich set of manipulation techniques to architecture design. This discrete perspective is not required however as Harada et al. (1995) have dealt with transitions between discrete and continuous manipulation of floor layout models.

In addition to the type of analytical frameworks adaptable to architecture design, prior work also demonstrates differences in scope chosen for an approach to the architecture design problem. While the scope is generally focused on the transition from function to form, the emphasis and granularity of previous approaches varies. A line of work from Chakrabarti (1994) is focused on configuration design in which alternative physical solutions are determined based on spatial constraints. This work develops a representation consisting of functional elements, generic objects, and standard objects. Generic objects are a basic class of several morphological solutions that satisfy a functional element while standard objects are a specific physical solution which satisfies its generic object (Liu et al., 1999). This representation facilitates progressive development of a design solution although the technique is not entirely practical for rapid implementation and is probably best reserved for computational tools which use the technique. In contrast, Allen and Carlson-Skalak (1998) developed a method that is less involved to implement. They developed a system function structure that is a variation of a traditional function structure. Here the system function structure defines the flow of energy, material, and signals among modules established from reverse engineering the product. This approach is attractive in some settings in particular because the representation format requires very little overhead in terms of putting the representation into practice.

A range of terminology and basic concepts is also evidenced in prior work. Erens and Verhulst (1997) decompose the architecture design phase into three domains: function, technology, and physical. Sander and Jantsch (1999) propose a skeleton map design model that uses a functional description as a skeleton for preliminary hardware layouts. Jensen (2000) addresses the spatial mapping of function to form by means of wirk elements, which are based on German design theory. A contribution of this wirk element approach is the capability of decomposing spatial regions and part surfaces that contribute to a given function or functions.

Each of the above references bring some useful concept to bear on the architecture problem. In addition to considering these previous engineering perspectives, this work also looks at the field of psychology. Many fields of study employ the use of models and representations. One approach is to investigate models from other fields such as mathematics, business, or economics. For purposes of getting to the roots of the problem, this work examines the field of cognitive psychology which is, among other things, in the business of understanding and explaining how people think in terms of representations. In addition to this perspective, it is useful to present key concepts from design and systems engineering to establish a reference that will aid in scoping the representation problem in the context of some larger design process.

2.3.1 Influences from Cognitive Psychology

The purpose of developing an architecture representation is to make architecture design easier. However, this representation venture has a potentially hazardous side effect. Hayes (1989) points out that the manner of representation will have a significant impact on the task. He refers to experiments where a given problem can be over an order of magnitude harder to solve depending on the problem representation. Norman (1983) also argues that one's perspective of the world and the tasks one is required to perform depends strongly on the representation used. While this variation in difficulty depends on the case at hand, it demonstrates how important the representation is to the task.

A set of requirements for architecture design are developed in a later section, but for now it suffices to present the architecture design task as a creative process to synthesize physical solutions based on design requirements and functional design information. The following sections develop the concepts used to develop the product architecture representation.

2.3.1.1 A Model for the Cognitive Process of Architecture Design

One useful model of creative cognitive processes such as architecture design is the Geneplore (Generation – Exploration) model developed by Finke, Ward, and Smith (1992). This model is based on the notion that creative activities generally follow a process shown in Figure 2.2.

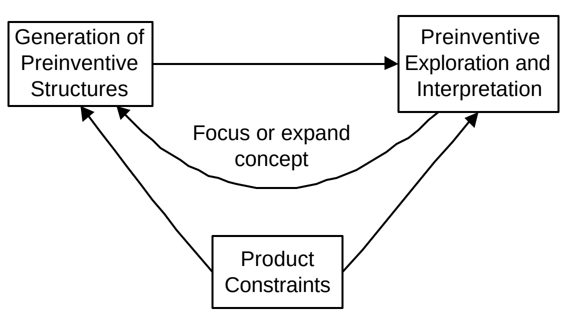


Figure 2.2 Geneplore structure

Finke, Ward, and Smith (1992) describe this model in terms of the processes, structures, properties, and constraints as given in Table 2.1. Each of these items is relevant to understanding the relation between human cognition and the design process. Benami and Jin (2002) use these concepts and the Geneplore stucture to develop a cognitive model of conceptual design. Since one objective in the present work is to generate a representation, the 'preinventive structures' is most relevant here.

Table 2.1 Genoplore model features (Finke, Ward, Smith, 1992)

Generative Processes	Exploratory Processes	Preinventive Properties	Product Constraints	Preinventive Structures
Retrieval	Attribute finding	Novelty	Product type	Visual patterns
Association	Conceptual interpretation	Ambiguity	Category	Object forms
Synthesis	Functional inference	Meaningfulness	Features	Mental blends
Transformation	Contextual shifting	Emergence	Functions	Category exemplars
Analogical transfer	Hypothesis testing	Incongruity	Components	Mental models
Categorical reduction	Searching for limitations	Divergence	Resources	Verbal combinations

The 'preinventive structures' heading indicates that several representation formats exist although this work focuses on one such structure as the foundation for the representation. The next sections develop this concept.

2.3.1.2 Representations in General

A representation can generally have four components as shown in Table 2.2. Markman (1999) discusses several levels of representations that are available in increasing levels of sophistication: spatial (includes some element of space), featural (property) based, network models (a graph – a set of connected elements), structured representations that may include visual, causal, and temporal information, and mental models (inclusion of future or potential representation states that are not in existence currently). The mental model concept will be explored because it appears as the most comprehensive and appropriate for the problem of representing physical systems in the context of an active design exercise.

Table 2.2 General Representation Components (Markman, 1999)

Component	Description
Represented world	The actual system domain
Representing world	The modeled system domain
Representing rules	Relations that map the represented world to the representing world
Representation Process	A method that uses the representation

2.3.1.3 Mental Models

The background for mental models stems from two main areas (Gentner, 1983). The first is cognitive psychology and those related areas that investigate the mind. Second, artificial intelligence research also provides theories about knowledge representation and processing. While different mental models have been developed to represent different problem domains, here the discussion focuses on mental models of physical systems to understand how a mental model can be used in architecture design.

Generally a mental model consists of three items: a set of objects that represent items of interest in the actual system, a set of relations and properties among the objects, and a notation for the objects (Markman, 1999). A simple drive train in Figure 2.3 serves as an example of these three items. The drawing itself is the notation. The objects are a base, two bearings, motor, shaft, pulley, belt, driven pulley, driven shaft, and some of the design variables as shown. Examples of two relations associated with the model are given in Table 2.3.

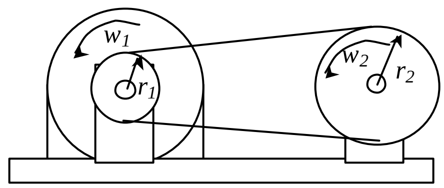


Figure 2.3 Example set of objects for a mental model of a physical system.

Table 2.3 Example set of relations for a mental model of a drive train.

w_2 will increase with a decrease in r_1
w_2 will increase with an increase in r_2
output torque will increase with motor torque

As the name suggests, mental models are observed, manipulated, and reasoned by the mind. Studies of mental models by Williams, Hollan, and Stevens (1983) have shown that people tend to reason about such models using relatively simple qualitative relationships such as those given in Table 2.3. White and Frederiksen (1990) argue that people think about physical systems with zero-order models, (whether or not an item is present or not), and first-order models (the direction of change of a parameter in the system). Additionally, White and Frederiksen indicate that quantitative relations in mental models are generally not used to mentally determine the degree of change. This suggests that it is reasonable to gear the architecture representation toward a format that facilitates this type of first order analysis and reasoning of the design. Knowledge of these human tendencies to utilize the models at a somewhat crude level of analysis offers a goal to seek during representation development.

2.4 DESIGN KNOWLEDGE

Chapter 5 treats the development of design guidelines and so it is useful here to present the distinction between product and process based knowledge. This distinction clarifies the kind of design knowledge being sought in this work.

2.4.1 Knowledge Types – Product versus Process

There are several models of the design process (Roozenburg and Eekels, 1995; Ullman, 1997; Otto and Wood, 2001; Pahl and Beitz, 1991; IEEE Computer Society) and sporadic references to the use of design knowledge throughout these model descriptions. Some work (Tomiyama et al., 2002; Reich, 2002; Frankenberger, et al., 1997) has made a clear distinction between two kinds of design knowledge: product based and process based. Product based information is considered to be knowledge regarding the manner of existence of a physical device or representation of such a device. This form of knowledge emphasizes the composition and specification of an object or artifact of design. Process knowledge, in contrast, refers to information about ***how*** the design process takes its course in terms of synthesis, analysis, or other activity attributed to a design technique, method, strategy, tactic, or other design operation.

Given the two broad types of design knowledge, there is a significant difference in how one may acquire this knowledge depending on which type is sought. Based on the reasonable premise that humans are the main agent in design processes (this excludes computer automated design processes which at this time are much less common), a study of process data requires the study of human behavior. Waldron and Waldron (1996) and Cross (1996) discuss methods suitable for such study in the context of attempting to better understand mechanical design. Several techniques are addressed including interviews, protocol analysis (verbal or think-aloud protocols and discussion protocols), a depositional method, case studies, a retrospective method, and process observation. The following shows why this work focuses on product based knowledge.

The motivation for studying product knowledge is based on a perspective well described by Frankenberger (1997): "Knowing the logic behind the make-up of technical systems, we should then be able to optimise the methods of making them." Roozenburg and Eekels (1995) also recognize this type of knowledge and refer to it as substantive knowledge. There are clear advantages to dealing with product based information. Compared to human behavior, product data is tangible, more available, and compliant when subject to examination. Due to the nature of product knowledge, it is clearly more accessible and observable than process data and this is the basis for focusing on only product knowledge in this study.

In particular, the information sought here is product based data that demonstrates good architecture design. Some criteria on the standards of quality for this data can be set now. Based on a concept shown in Figure 2.4, there are multiple levels of sophistication, which describe the product data (Cohen, 1995).

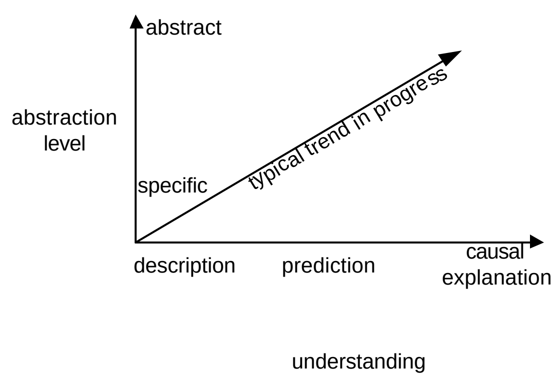


Figure 2.4 Model of scientific progress (Cohen, 1995)

Minimally, the data for good architecture design should accurately relate good architecture design in terms of some relevant architecture parameters. This is consistent with Cohen's prediction level and although this prediction does not require an explanation about why such a state of parameters typically results in good design, the data simply correlates a set of conditions with a frequent effect associated with that condition. If an explanation, which is desirable, is available then the data would provide evidence to support a hypothesized causal effect.

This scenario would match Cohen's most sophisticated level of understanding. While causal explanations are certainly desirable, success of this research is fulfilled even at the prediction level of explanation provided that the predictions are reasonably novel and useful to the problem of architecture design. In short, the kind of data sought is new evidence that shows what composition of architecture results in good design.

A second specification for the type of knowledge sought is also indicated on Cohen's diagram–the degree of generalization. In the case of product based knowledge, the data should span a reasonable set of systems that are considered common in mechanical design. Several metrics could be imposed here to rigorously define the type of design or low-level parameters such as number of parts, cost, etc (Shenhar, 1998; Felligi et al., 1997). Instead, the scope of systems under investigation is bounded by a more roughly defined constraint that biases the data set toward systems that are mainly mechanical (excludes circuit board systems for example) and are generally at the same scale of consumer products (excludes the really big and really small).

Other distinctions in terms of design knowledge can be made beyond those of process versus product based knowledge. Nowack (1997) presents a thorough review of different knowledge definitions and classification schemes. The lack of a common format or widely accepted set of knowledge types presents a ripe classification problem that is approachable possibly with the application of taxonomy methods (Dunn, 1982). Classifying design knowledge is outside this study, but a brief review of selected knowledge types serves to specify, in terms of function, the knowledge deliverable of this work.

2.5 SUMMARY

This chapter presents details of prior work that form the foundation for this document. The following key items are given:

- A research approach used in the next three chapters,

- A perspective of prior methods to further demonstrate the need for an improved architecture design method,
- A review of prior work to develop representations related to architecture design such as conceptual design modeling schemes,
- A review of relevant work in cognitive psychology related to knowledge representation and mental models in particular,
- A detailed view of design theory fundamentals including the terminology of different design phases and activities,
- A discussion of design knowledge with respect to guideline development including the distinction between product based knowledge and process based knowledge.

Chapter 3 – Representation

The purpose of this chapter is to present a formal representation scheme for product architecture. A function structure is a successful example of how an abstract product description can be represented in a relatively well-defined and repeatable manner (Kurfman et al., 2000). One downside to the function structure concept is that it presents the designer with a huge dissociation between function and form. There is no well-defined middle ground between function and form. There is no concrete notion of how to conceptualize, represent, model, or much less manipulate this transition phase toward some desired outcome. This chapter presents a conceptual framework for this transition state between function and form in order to impose some structure to the currently ill-defined design space of product architecture.

3.1 Introduction and Objectives

In adhering to the research approach outlined in Figure 2.1, the first task is to make observations based on the current needs of industry. Specific industry needs will be detailed in the next section, but here the immediate discussion highlights the main gaps in current research. Several observations can be made based on the prior work presented in Chapters 1 and 2. Note first that some researchers place emphasis on developing a representation to suit a narrowly focused need. For example, Jensen's (2000) work develops the wirk element concept to explain function integration while Erens and Verhulst (1997) prescribe a representation to address architectures for product families. In other cases, methods such as the Pahl and Beitz (1996) technique feature a relatively broad approach that captures both function and form. However, these methods generally are lacking in detail and capacity to effectively represent architecture issues in great depth. In terms of the narrowly focused methods, they typically address the architecture design problem at one of multiple stages of the design process. For example, some work emphasizes techniques that treat architecture during functional modeling while others begin even earlier with methods to identify the most appropriate architecture very early in the design based primarily on customer needs information (Stone, 1997; Zamirowski and Otto, 1999). In addition, there is a great deal of variation regarding the granularity and resolution of architecture representations. For example, Allen and Carlson-Skalak (1998) do not represent as much functional detail as Chakrabarti's (1994) work. Another observation is that there is a significant variation in the overhead required to implement previously proposed representations in a design method. Some representations are inherently more computationally intensive than others. Given these observations, there is an opportunity to develop a representation that improves upon the capabilities of existing techniques in terms of being practical to implement and highly capable for handling the architecture design problem.

The premise is that an architecture representation offers designers benefits if it presents a well-defined domain in which designs can be documented, observed, and manipulated. Specifically, such a representation is expected to improve upon previous representations with respect to those three activities. The primary objective is therefore to generate a formal architecture design representation that fulfills the criteria above. In order to reach this goal the following actions form the research plan which is consistent with the overall research approach:

1. Define the requirements and criteria for success of an architecture representation.
2. Develop a formal representation for product architecture.
3. Test the representation against success criteria.

3.2 REQUIREMENTS

The current task is to understand the nature of the architecture problem in order to develop requirements for the representation. Upon reflection of the architecture design process, five major characteristics are found as shown in Table 3.1. Of course, many other factors such as safety and reliability are relevant but are assumed to be addressed via the set of customer needs developed early in the design.

Table 3.1 Architecture Design Characteristics

Iterative
Successive approximation
Large number of relevant issues to consider
Strong interdependence on multiple factors
Non-constant starting point
Concurrent engineering
Constraints on the order of operations and scheduling issues

Like other phases of the design process, architecture design exhibits an iterative aspect in the solution search. Secondly, one of the prime difficulties of architecture design is the overwhelming number of issues the designer must consider when transforming from function to form. In addition to the volume of issues to consider, there is generally a high degree of coupling among them and so keeping track of these dependencies is a problem. Not all design projects have the same starting point and many projects involve legacy artifacts from prior work that dictate the departure points for design activities. While often simply referred to as the transformation of function to form, architecture design involves simultaneous activities or concurrent engineering such as developing aesthetics, user interfaces, exploring alternative concepts, exploiting promising concepts, improving efficiency, reducing number of parts, addressing manufacturing concerns, etc.

It is clear that architecture design is a complicated process and that several requirements are needed to develop a successful representation. After considering the nature of the architecture problem and the characteristics that typify the challenges of architecture design, a few requirements are generated and listed in Table 3.2. Although these requirements are developed for the architecture design phase, they are generally compatible with representations used in other aspects of design.

Table 3.2 Representation Requirements

Accommodate concurrent engineering
Reconcile short term memory capacity with the huge set of design issues
Do no harm
Support multiple start points and iteration
Support generation of multiple alternatives
Be formally defined
Be practical to implement
Be robust to typical design project noise
Facilitate progressive application of device partitions
Facilitate efficient management of the design

The representation must support common design processes where multiple tasks are performed in parallel. For example, the representation should not preclude concurrent external interface definition and concept development for functional modules. Whether explicitly or implicitly, the representation must give rise to a relatively large set of relevant architecture design issues in a coherent format that keeps these issues at the forefront of the designers thought. The reason for this need is to mitigate the limitations of a designer's short-term memory which hampers full and frequent consideration of multiple design factors. Do ***no*** harm should not be taken literally, however. The representation must generally improve the design situation relative to not using any representation at all. The representation must allow for alternative design starting points as well as the revisions that occur during iteration among different phases of the design process. Additionally, the representation must facilitate diverging design activities in which several alternatives are generated. In order to improve operational repeatability and future extensibility, the representation must be well-defined. Low overhead and resources should be required so that the representation is practical to implement. The representation must work despite reasonable variations in designer skill, product domain, project type, and designer resources. A means for establishing product partitions in terms of modules and components should be given in the representation but the representation should not require such a partition or selection of components to be imposed from the starting point. The representation must allow the designer to control the design process from both strategic and tactical perspectives. That is, the representation must include observable and controllable information relevant to both high level (strategic) product management needs and lower level (tactical) product development processes. This promotes utility and reuse of the representation across multiple users in the product development cycle.

Given the above requirements and the background discussed earlier in Chapter 2, the problem of representation development is clarified. No assumptions are placed on the type of design problem other than the product is mainly mechanical in nature and the scale is somewhat restricted. That is, the representation is not intended to be applicable to electronic circuit design or software design although electronics are not entirely ruled out since mechatronic devices are applicable. While not fundamentally limited to a particular scale of device, the representation is directed mainly toward the small and medium scale of devices like mechanical pencils, power tools, washing machines, etc. This scale excludes the very small such as MEMS devices and the very large such as cars and aircraft.

3.3 REPRESENTATION DEVELOPMENT

The mental model concept developed above is the basis for the representation developed here. Figure 3.1 illustrates three items derived from the mental model structure: the lexicon, notation, and operations on the notation. These items are shown in the context of design issues and design variables related to architecture design. Each item in the representation is defined in Table 3.3. The remainder of this chapter develops the lexicon and notation sets while the operations set is developed as guidelines and a method in the next two chapters.

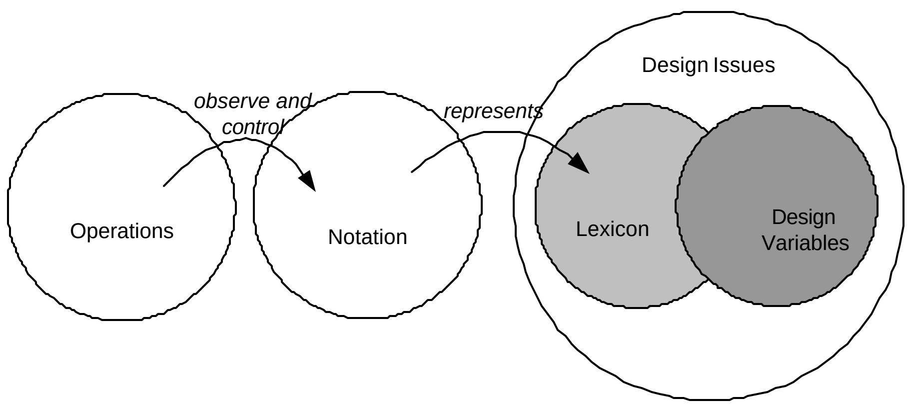


Figure 3.1 Architecture Design Representation

Table 3.3 Representation Definitions

Term	Description	Examples
Design Issues	Set of all design variables relevant to architecture design	Modularity, serviceability,
Design Variables	Set of controllable design variables	Size, shape, position, orientation, mass, etc.
Lexicon	Set of design variables represented	Functions, modules
Notation	Set of observable and controllable objects that represent the lexicon	Sketch, mathematical expression
Operations	Set of operations that manipulate design variables	Design actions – synthesis, analysis, evaluation, etc.

Given this framework, the research task is to develop the lexicon and notation shown in Figure 3.1 of the mental model framework.

3.3.1 Lexicon Development

The lexicon is generated by first searching for a relatively large set of issues that are relevant to architecture design. Following this search, the list is pruned to yield the lexicon. It is well known that product architecture is related to a vast number of design issues ranging from aesthetics to manufacturing choices (Cutherell, 1996). By performing an individual and group brainstorming search as well as a literature search, a list is generated that contains over 200 generally different issues. This list is given in Appendix B. Nearly every design issue is directly or indirectly related to some design variable although there are a few exceptions. For example, patent infringement problems are certainly design issues, although one is hard pressed to define a succinct and meaningful set of design variables that reflect patent infringement issues.

The lexicon is refined by considering each design issue and developing a manageable set of relatively important, observable, and controllable artifacts of the design. The rationale is based on the Pareto Principle (the 80/20 rule) and so the goal is to identify those relatively important issues and design variables with the reasonable expectation that this reduced set captures the bulk of what is really important in product architecture. Table 3.4 presents the derived lexicon, which is small compared to the number of design issues considered. Criteria for selecting this final lexicon are related to two issues. First, the terms in the lexicon are chosen because they are somewhat abstract in the sense that each term generally subsumes a set of terms that are less abstract. Secondly, the lexicon does not include those items that are outliers in terms of relevance with respect to architecture design. One concern about lexicon pruning is about the potential for restricting designer creativity. However, any detrimental impact on creativity should be minimal given that the design space is not restricted although the designer's attention is more in tune with a restricted set of design issues. The following discussion addresses the relation between creativity and restrictions on the terms in the lexicon.

Engineering creativity can be defined by the characteristics of the design solution. A design is creative if it is original. Based on empirical studies of creativity, Finke (1992) has found that certain restrictions on the objects used to design will in some cases enhance creativity and in other cases detract from creativity. If the objects or building blocks for a designer restrict the set of solutions relative to the solutions in the design space of interest, then creativity will be potentially hampered. If, however, the restrictions merely force the designer to view the design problem from a perspective outside the norm of his or her view, then creativity will likely be promoted.

The effect in this second case is offering the designer potentially new search directions by avoiding, or restricting, the normal cues that lead a designer to solutions typically associated with a given perspective. The difficulty is in contriving a restriction that forces a designer to think outside the box while not overly restricting the design space. In the case of pruning the design issues to a smaller list as given in Table 3.4, this restriction forces the designer to focus more intently on a relatively small set of issues that normally might get lost in an unrestricted clutter of design issues. The next task is to generate a proxy or notation for modeling the lexicon.

Table 3.4 Architecture Lexicon

TERM	DEFINITION	CONSTITUENTS and STRUCTURE
Function	A form independent operation that a machine imposes on a set of energy, material, and signal flows.	Constituents: Form independent operation: a verb-noun (operation-flow) phrase such as "Dissipate Heat" A set of input flows of energy, material, and signals A set of output flows of energy, material, and signals Structure: The form independent operation acts on the input flows to yield the output flows.
Industrial Design	The rules and issues governing the physical	Constituents: A set of gestalts: physical effects and their relation to the human

Syntactics	embodiment with respect to human perception of the embodiment.	perception of those physical effects Structure: The gestalts are mapped to a physical solution to characterize the physical solution in terms of human perception
Device Operations and User Activities	The operations and activities that are imposed on the device during the device lifetime.	Constituents: A set of user activities associated with product operation. Structure: The user activities are given with respect to the device and the physical flows. The activities are related among themselves temporally.
Customer Needs	Explicit and latent needs and desires of the customer.	Constituents: A set of needs. Structure: The needs are generally ranked in order of importance.
Physical Solution Topology	A physical embodiment of some set of functions. (A concept in the physical or form domain)	Constituents: A set of physical embodiments that satisfy some function and the connectedness of those embodiments. Structure: The physical embodiments are given and related among each other in terms of geometry and material. In architecture design, the physical solutions are generally only roughly sized in terms of relative size, shape, position, and orientation. Physical solutions are partitioned into physical modules and components.
Function to Form Mapping	The relationship between a set of functions and the physical embodiment that instantiates that functionality.	Constituents: A set of relationships that map a set of functions to a set of physical embodiments. Structure: Functionality is correlated with spatial regions of the product. (Similar to the wirk element concept (Jensen))
Manufacturing Choices	The specifications for how a subset of a physical solution is manufactured.	Constituents: A specification that describes the choice of manufacturing for one or more components. This choice generally distinguishes between OEM and custom fabricated parts and can further specify those custom fabrication techniques. For example, injection molding is such a technique. Structure: Specifications are correlated to components, physical modules, and possibly functional modules.
Component	A physical part.	Constituents: Geometry and material specifications that describe the part. Structure: A description of the geometry and material specifications either explicitly or more vaguely by referring to a common part name. (A set of physical parts is enumerated in Greer, et al. 2002).
Functional module	A set of related functions.	Constituents: A set of functions. Structure: A functional module may contain both functions and lower level functional modules.
Physical module	A set of components that exist as a stable assembly even without any external effort holding the items together. (Greer, 2002)	Constituents: A set of components. Structure: A collection of components and possibly physical modules in terms of a spatial region.
Flows	A set of energies, materials, and signals that are processed by the product. (Enumerated by Stone, et al., 1999).	Constituents: A set of energy, material, and signals. Structure: Flows can be defined both non-spatially with respect to functions and spatially with respect to physical solutions.

Interfaces	The physical regions where physical flows exist (not including the regions internal to component material).	Constituents: A set of spatial regions where energy and / or material flow between components or between a component and the external environment. Structure: Physically, interfaces are the partitions within a physical solution and they are given spatially. Functionally, interfaces are equivalent to the flows between functions.
Functional Solution Topology	A set of functions and flows that satisfies some overall main product function. (A concept in the functional or form-independent domain).	Constituents: A set of functions, flows, and their connected arrangement. Structure: The functions and flows are given so that their arrangement represents the processing of flows from device input to device output.
Product family elements	The family platform, common components, similar components, and distinct variant components.	Constituents: Generally two sets: common and uncommon components. Common components makeup the platform while unshared components support different variants. Structure: Common components are distinguished from unshared components in order to identify the platform.
Relative motion	The degrees of freedom and range of motion for components and physical modules.	Constituents: A degree of freedom and range of motion specification for a component. In cases of compliant mechanisms, this specification may be given relative to itself rather than a fixed frame. Structure: In most cases, the relative motion specification is given relative to some fixed frame although in the case of compliant mechanisms, this specification can be given relative to the component itself. (Refer to R-links and C-links in Greer, 2002).

3.3.2 Notation Development

The cockpit for a modern aircraft is relatively sophisticated in terms of fusing vastly different informational items from multiple time varying inputs. These inputs as well as output interfaces are presented to the pilot in an intuitive, coherent, and relatively complete format in order to improve the pilot's effectiveness in observing and controlling the flight situation. This cockpit–pilot interface is very analogous to the notation–designer interface. The notation is only an external proxy for the design representation. However, this proxy has great influence over the designer's ability to design effectively just as the objects in the lexicon influence the designer.

3.3.2.1 Notation Overview

The requirements developed earlier and presented in Table 3.2 are directly applicable to requirements for the notation. In a broad view of the architecture problem and the lexicon in particular, it is clear that the notation must facilitate the representation of a wide range of design artifacts as indicated by the lexicon content.

An assumption for notation development is that the notation must incorporate all elements of the lexicon either as an input to the notation or as part of the notation itself. Of course, many tools already exist for representing many of these items. This research leverages the utility of these existing notational devices and creates new notational devices where prior work falls short. Finding these shortcomings in previous work is the next discussion.

By comparing elements of the lexicon shown in Table 3.4 to those activities in the design process shown in Figure 1.1, one can see where elements of the lexicon come into play. In addition, Table 3.5 below correlates existing representations to phases in the design process. It is clear from Table 3.5 that several existing notational devices are already developed. The table also indicates a weakness in the Concept Layout Generation phase where an improved notation for the layout is needed.

Table 3.5 Current notations in design

Design Task	Existing Notations
Problem Clarification & Definition	Customer needs analysis results Mission statement Benchmarking QFD
Industrial Design	External illustrations Foam models
Functional Modeling	Black box Function structure with functional models identified
Physical Solution Generation	Morphological matrix of schematic physical solutions Solution description in textual format
Physical Solution Combination – Concept Layout Generation	Rough geometric schematic ***Architecture notation needed here*** Mathematical models Proof of concept physical prototypes
Manufacturing and Assembly Design	Manufacturing data, selected processes, source of material Bill of materials Assembly tree Force flow diagram
Final Form Specification	CAD, solid models with complete specifications on all parts

Upon inspection of the above table, notational devices used for design are rich in terms of their visual content and this has advantages based on work from Kremer (1998) who presents several reasons for using visual languages as opposed to relatively linear text: Abstract reasoning is pictorial in nature and has two and three dimensional aspects and so it is logical that visual reasoning may be more efficient than linear verbal language. Visual organizations are efficient for chunking information and thus mitigate short-term memory limitations. Thought operations may be transformations of images and images are analogous to tasks, while linear languages generally bear no analogous relation to the task they describe. For all these reasons, a highly graphical notation is adopted in this work.

Two main factors drive the representation notation. First, the notation is limited to a maximum of seven chunks of information in order to satisfy short-term memory constraints. Secondly, the notation must include elements of the lexicon in a meaningful and accessible format. Results for the notation development consist of six diagrams listed in Table 3.6. The inputs required to generate the diagram are shown.

Table 3.6 Architecture Notation

Layout Diagram	Description	Minimum Inputs	Outputs
Spatial Constraints Diagram	A diagram mapping spatial constraints to spatial regions	Customer needs, Requirements, Functional Model, One Physical Solution choice	Spatial constraints External Physical interfaces
Function Layout Diagram	A diagram mapping functions to spatial regions	Spatial Constraints Diagram Physical product (if redesign)	Function to form mapping Candidate physical modules and partition choices
Physical Solution Diagram	A conceptual physical solution for a subset of the spatial diagram	Function Layout Diagram Alternative physical solution choices	Spatial layout of physical solution Relative motion
Partition Diagram	A tree structure	Physical Solution Diagram	Physical partitioning into modules and components Assembly choices
Manufacturing Diagram	A diagram mapping manufacturing choices to spatial regions	Physical Solution Diagram	Manufacturing choices
Product Family Diagram	A diagram mapping product family elements to spatial regions	Function Layout Diagram	Product family elements

Each of the six notations is defined below and includes a set of nomenclature and a procedure for specifying each notation. The procedures are mainly geared for an original design although the redesign case can be handled using the same procedures by simply including appropriate constraints that follow from the existing device. A Skil Twist power screwdriver and a Swingline stapler are used as example cases to illustrate the six notation diagrams. The following definitions are used in the notation:

Product space: The spatial volume bounded by a product.

Region: A closed volume including the interior and edges. A region can be either the product space or a subset of the product space.

Spatial Diagram: A notation that represents the product space overall or regions of material in the product space. The diagram may be realized with a sketch, solid model, or appropriate data structure.

3.3.2.2 Spatial Constraints Diagram

The purpose of the spatial constraints representation is to show geometric constraints of the product overall. Supporting concepts for this notation include industrial design themes since these have a high content of spatial information. Table 3.7 provides the general nomenclature, and Figure 3.2 shows the spatial constraints diagram for the Skil Twist screwdriver. The diagram is a silhouette of the product much like a control volume. External flows show the main interactions with the environment and the dimensions give some idea about the scale and known constraints.

Definition: The spatial constraints diagram consists of a roughly sized product boundary, all external energy, material, and signal flows oriented in space, and known geometric constraints specified.

Table 3.7 Spatial Constraints Nomenclature

Notation	Definition	Rule Set
→	Energy, Material, or Signal flows external to the product boundary	Show flows relative to the product boundary in terms of position and orientation Flows are equivalent to the flows on the black box diagram
□	Boundary of some product region.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region Multiple boundaries of regions collectively form the product boundary
Text	Dimensions or descriptions of geometric constraints	Spatial information shown where relevant

Diagram Generation Procedure:

1. Reproduce the functional black box.
2. In order of importance, spatially orient the energy, material, and signal flows based on requirements information, functional information, and the currently selected set of physical solutions.
3. Establish a product boundary by reshaping the black box boundary according to the magnitude and direction of the energy flows and the size, shape, type, and amount of the material flows and their relative location.
4. Identify and label dimensional constraints.

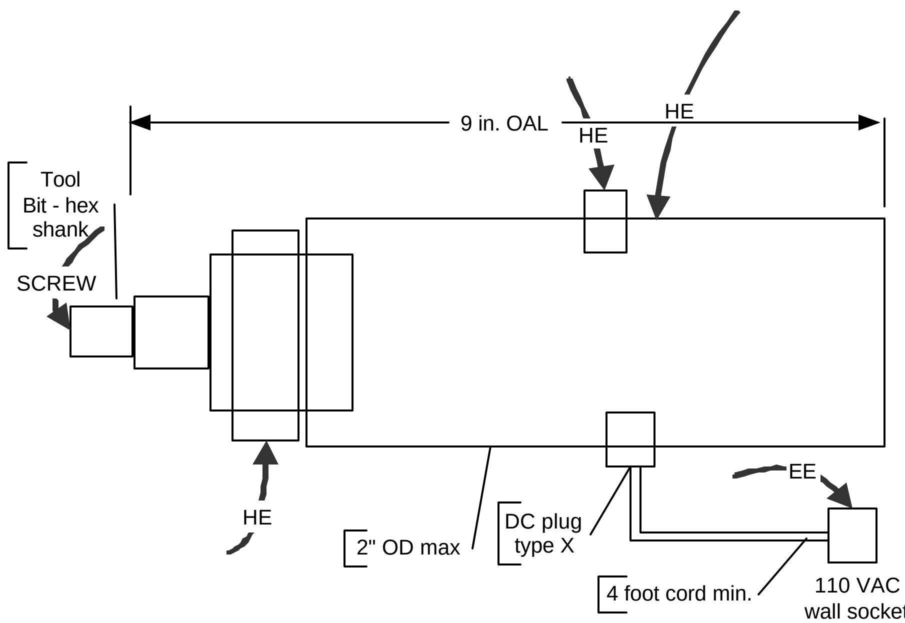


Figure 3.2 Spatial Constraints Diagram for a cordless screwdriver

3.3.2.3 Function Layout Diagram

The function layout is based somewhat on the wirk element concept (Jensen, 2000) in that spatial regions are associated with some functionality. The main purpose of this notation is to show the basic internal functions and flows in a spatial format. Table 3.8 gives the nomenclature, and an example of a function layout is given in Figure 3.3. One outcome of this particular notation is the ability to predict candidate physical modules in a manner not previously reported. This is covered in a section later in this chapter.

<u>Definition</u>: The function layout diagram consists of a product boundary, all flows external and internal to the device, layout elements, and layout intersections.

Table 3.8 Function Layout Nomenclature

Notation	Definition	Rule Set
	Energy, Material, or Signal flows external and internal to the product boundary	Show flows relative to the product boundary in terms of position and orientation Show flows from input to output along the same function path as the function structure
	Layout element – a region that corresponds to some function	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
	Layout intersection - A region that is a unique intersection of two or more layout elements.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
Text	Flow and function descriptions	Function information is shown for each layout element and layout intersection All functions from the function structure should be identified

<u>Diagram Generation Procedure</u>:

1. Reproduce the product boundary based on the spatial constraints diagram and include the flows but do not include dimensions.
2. In order of flow importance, establish regions for each functional module first and then each function while maintaining the same functional topology from the function structure.

a. Establish a layout intersection (dashed line) for each functional module identified in the function structure. Module identification in the function structure is described by Stone (1997).
b. Establish a layout element (solid line) for each function in the function structure.
c. Establish internal flows that connect layout elements and layout intersections.
d. Size the layout elements and intersections based on the physical solutions size, the magnitude and direction of the energy flows, and the size, shape, type, and amount of the material flows and their relative location.

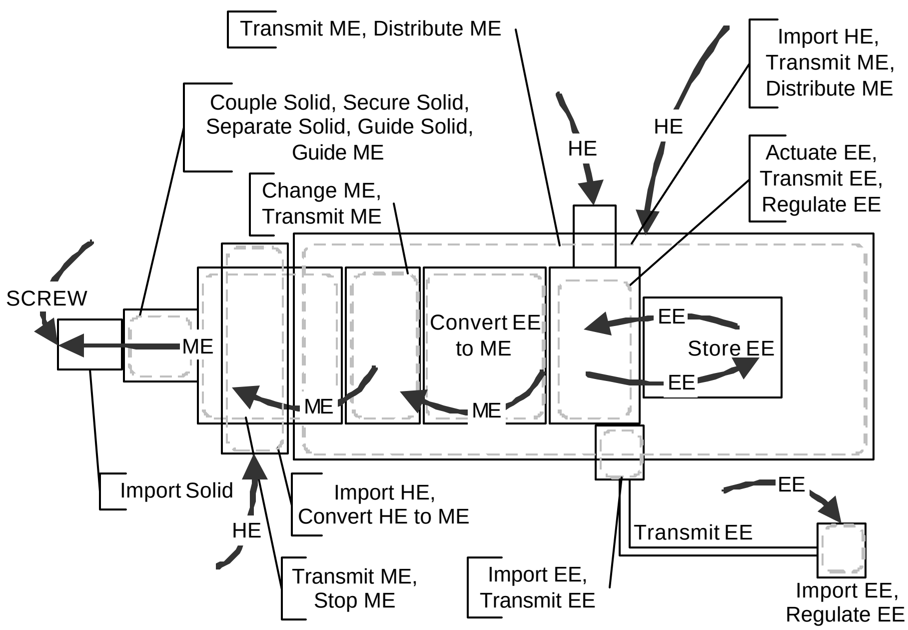


Figure 3.3 Function Layout Diagram for a cordless Screwdriver

3.3.2.4 Physical Solution Diagram

The physical solution diagram is intended to address the choice of morphological solution (Zwicky, 1948) for a set of functions and the relative spatial arrangement of those solutions. This particular notation is derived in part from Chakrabarti's (1994) work in that relative motions are shown. Table 3.9 provides the nomenclature, and an example is given in Figure 3.4. Note that gross motions are given while some, such as the motion of the planetary gear set, are not. One reasonable threshold for selecting those motions to include is to consider how significant the motion is to the physical solution. This allows the designer to be flexible when including physical motions.

> Definition: The physical solution diagram consists of a product boundary, physical solution descriptions (text, schematic, or sketch) and arrows indicating motion of parts.

Table 3.9 Physical Solution Nomenclature

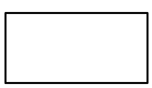
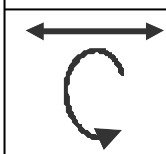
Notation	Definition	Rule Set
	Layout element – a region that corresponds to some function	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
	Layout intersection - A region that is a unique intersection of two or more layout elements.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
	Relative motion of a region	Indicates a significant relative motion corresponding to a physical solution at some region May generally be translation or rotation
Text	Physical solution descriptions	Indicates one or more physical solutions that are consistent with a spatial region Alternative physical solutions can be indicated on the same layout provided that both solutions share approximately the same geometric specifications

Diagram Generation Procedure:

1. Identify a layout element or layout intersection.
2. Indicate (with text, schematic, or sketch) the physical solution that performs the functionality of that layout element or layout intersection.
3. Repeat steps 1 and 2 for all layout elements and layout intersections.
4. Identify regions in the device that exhibit relative motion.
5. Label those relative motions using an arrow to indicate direction.

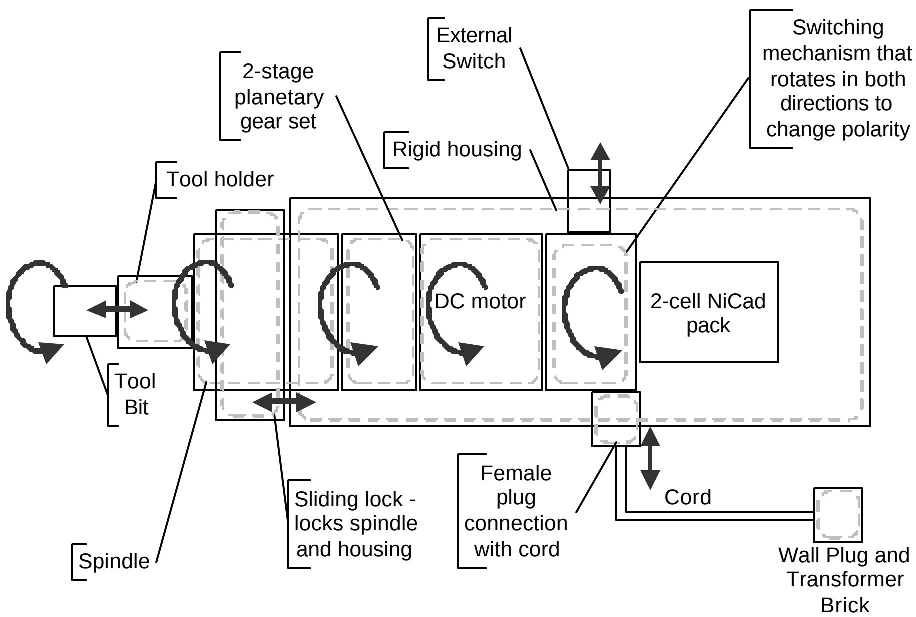


Figure 3.4 Physical Solution Diagram for a cordless screwdriver

In this power screwdriver diagram, one can clearly illustrate multiple options for different functional elements and intersections. For example, one may choose to include alternatives to the 2-cell NiCad pack. A second leader with description can document these alternatives to include such items as a lithium ion battery or alkaline battery.

3.3.2.5 Partition Diagram

The purpose of the partition scheme is to establish physical modules, components, and the manner in which they are connected. This notation is based on the Design Structure Matrix (DSM), and an extension of the DSM called the branch diagram, which is a tree structure. Nomenclature for both of these items is given in Table 3.10 and background on the development of the branch diagram is given in (Van Wie et al., 2001).

Definition: The partition diagram consists of the hierarchical relations among modules, components, and their physical interfaces.

Table 3.10 Partition Nomenclature

Notation	Definition	Rule Set
Design Structure Matrix (DSM)	A *component – component* matrix showing the relation between components and interfaces	Each element can indicate the existence of an interface (1) or not (0) Each off-diagonal element represents an interface between two components Each diagonal element represents an interface between a component and the environment external to the device Boxes within the matrix indicate a physical module among those components
Branch Diagram	A tree structure showing the hierarchy among physical modules and components	The main parent node is the product and subsequent lower levels are physical modules and components subsumed according to the DSM Squares represent physical modules and these correspond 1 to 1 with the physical modules in the DSM Circles represent components

Diagram Generation Procedure:

1. Partition regions of the device physical solution diagram into physical modules and components.
2. Generate a DSM including physical interfaces among components based on the partitioning scheme.
3. Generate a branch diagram based on the DSM.

	base	staple die	open lever	die spring	die pin	flat back plate	btm pad	arm lower	staple guide	staple follower	front brace	upper housing	upper cap	upper spring	upper arm	space	arm pin	follower spring
base	1																	
staple die	1	1																
open lever	1		1															
die spring	1																	
die pin	1																	
flat back plate				1	1	1												
btm pad	1						1											
arm lower	1	1						1										
staple guide		1						1										
staple follower								1	1									
front brace		1						1										
upper housing												1						
upper cap												1	1					
upper spring								1				1	1					
upper arm														1				
spacer															1			
arm pin	1							1				1			1	1		
follower spring										1					1		1	

Figure 3.5 Design structure matrix for a Swingline stapler

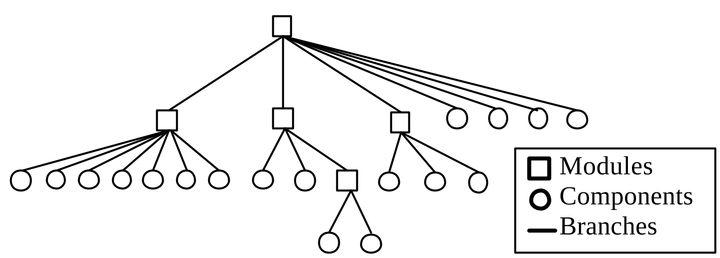


Figure 3.6 Partition Diagram of the same Swingline stapler

3.3.2.6 Manufacturing Diagram

The purpose of this diagram is to establish the manufacturing choices for components and modules. The primary format for this notation is a spatial diagram although other representations such as a branched diagram or bill of materials could potentially be substituted depending on the particular emphasis of the designer. The nomenclature as shown in Table 3.11 is reused to some degree from the previous diagrams.

> Definition: The manufacturing diagram consists of manufacturing choices such as material, processing, in-house sourcing, or OEM sourcing associated with a spatial region.

Table 3.11 Manufacturing Nomenclature

Notation	Definition	Rule Set
	Layout element – a region that corresponds to some function	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
	Layout intersection - A region that is a unique intersection of two or more layout elements.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
Text	Descriptions of manufacturing choices	Indicates the choice manufacturing with respect to a product region

Diagram Generation Procedure:

1. Identify the material and manufacturing choice for each region or component or module.
2. Identify in-house or OEM sourcing.

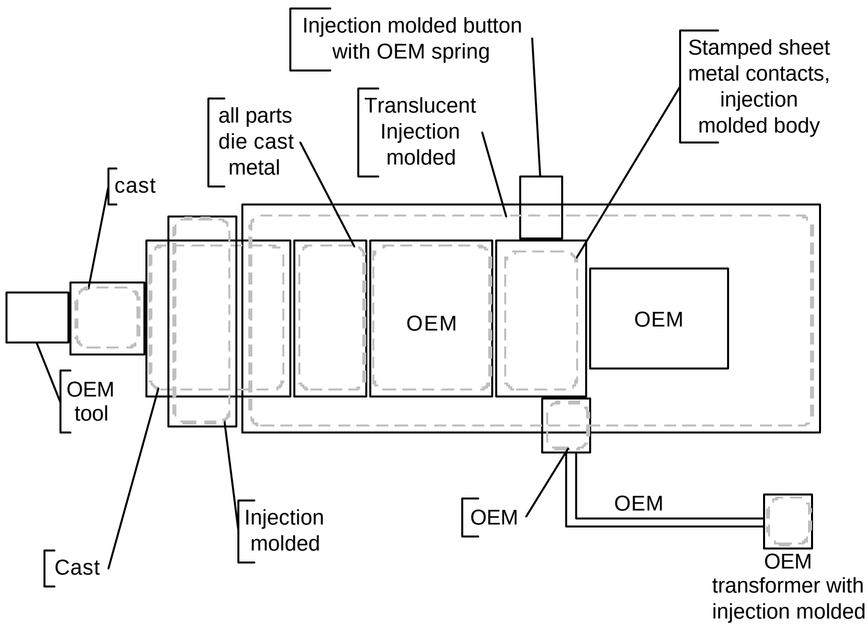


Figure 3.7 Manufacturing Diagram

3.3.2.7 Product Family Diagram

The purpose of the product family diagram is to highlight those regions or parts of the product that are common, similar, or different than others in the product family. Nomenclature for this diagram is shown in Table 3.12. Note the distinction between common and cousin regions which is based on work from (Kleespies, 2002). The following figure provides a hypothesized estimate of those items likely to be common among other (currently nonexistent) variants of the Skil Twist family.

> <u>Definition</u>: The product family diagram consists of common, similar, and unique items among a product family with respect to the current device being designed. These items may include functions, modules, or components.

Table 3.12 Product Family Nomenclature

Notation	Definition	Rule Set
	Layout element – a region that corresponds to some function	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
	Layout intersection - A region that is a unique intersection of two or more layout elements.	Position and orient according to approximate geometry of a selected set of physical solutions (the current physical solution topology) Boundary of a region is sized according to the energy and material throughput at that region
	A region common among other variants in the family portfolio	Used when a region is common to a region from one or more other variants in the family portfolio
	Region that is cousin (similar, but not unique nor drastically different) with a region from other variants in the family portfolio	Used when a region is similar to a region from one or more other variants in the family portfolio
Text	Common or cousin region descriptions	Indicates the nature of the region that is common or cousin – such a description may be in terms of either functionality ***or*** the physical solution

Diagram Generation Procedure:

1. Identify common parts and modules or identify common layout elements and layout intersections.
2. Identify similar parts and modules or identify similar layout elements and layout intersections.

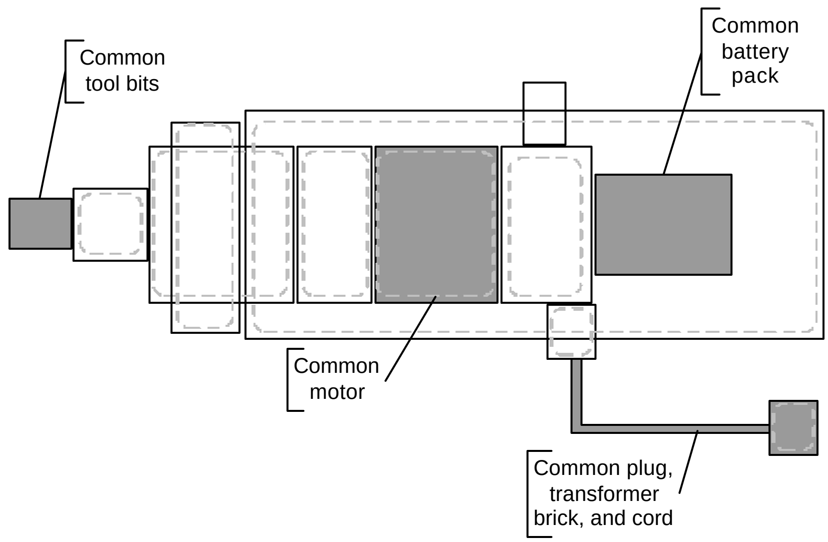


Figure 3.8 Product Family Diagram

3.3.3 Representation Discussion and Assessment

The overall representation scheme is a six element network that evolves as the design progresses based on design iteration. It relies heavily on the use of a spatial diagram which acts as a skeleton for separate but related architecture diagrams. Each notation diagram supports the others due to the inherent coupling among design issues. Any representation diagram can be used as an initial starting point given that minimal input requirements to that representation are met. By using the input minimums from Table 3.6 shown earlier, Figure 3.9 shows a logical approach for creating an initial representation. Note that the input information such as physical solutions in a morphological matrix may not all be technically feasible since activities used in generating a morphological matrix, such as brainstorming, often diverge far from feasible solutions in the search for novel ideas. The designer should therefore be cautious when using such input information in architecture design. The dotted lines in Figure 3.9 indicate optional or alternative paths.

One potential alternative not shown in the Figure 3.9 is the option of placing more initial emphasis on the shape / size of physical modules and building upon these initial conditions to form the spatial constraints diagram. The advantage of this option is the potential for more systematically shaping the spatial constraints diagram based on the shapes of physical modules. Exploring this approach is left to future work.

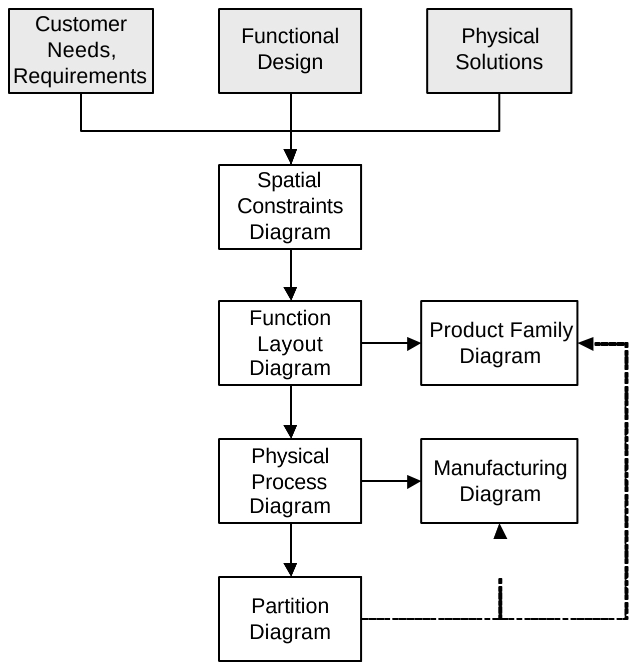


Figure 3.9 Representation initiation sequence using minimum inputs.

For purposes of evaluating the lexicon and notation, the developed representation is compared against the requirements in Table 3.13 below.

Table 3.13 Representation Assessment with respect to Requirements

Accommodate concurrent engineering	The representation does support parallel development of each of the six notations given that the minimum inputs for each notation are known.
Reconcile short term memory capacity with the huge set of design issues	The representation is chunked into only six units of information yet the lexicon is completely addressed in the notation.
Do no harm	The representation should enhance creativity since the representation is restrictive only at an abstract level based on the limited, but sufficient lexicon.
Support multiple start points and iteration	The representation allows the designer to explicitly impose constraints according using the most relevant notation diagram in any order the designer chooses provided that the minimum inputs for each notation are known.
Support generation of multiple alternatives	The representation does facilitate documentation of alternative architecture solutions.
Be formally defined	The representation is defined precisely to support repeatable application of the representation.
Be practical to implement	The representation does not require excessive overhead to implement. One can develop a full representation with pencil and paper without excessive burden.
Be robust to typical design project noise	The representation is flexible to product type and designer experience level.
Facilitate progressive design	The representation allows concepts to be developed without undue and inflexible constraints imposed from the outset.
Facilitate efficient management of the design	The representation facilitates project management by providing information at both high and low levels of detail.

The representation supports all of the requirements in a practical and comprehensive manner and to extend the assessment, the following section illustrates the representation utility through specific design applications.

3.4 APPLICATIONS

While the above brief evaluation of the representation with respect to the prescribed requirements is a logical test, it is also useful to examine the influence of the representation on applications. Based on the premise that one central design activity in general is synthesis, the impact of the representation is shown in terms of a two specific example applications that are relevant to synthesis: designing for modularity and designing for flexibility.

3.4.1 Designing for Modularity

Modularity is about the partitioning of a design and is addressed in terms of both function and form. The modularity issue plays a role in both product architecture and portfolio architecture. Here the focus is on product architecture although the representation does also support the design of product families to some extent. While multiple types of modularity exist such as sharing, swapping, bus, sectional, and mix (Otto and Wood, 2001), the primary issue is the distinction between integral and modular. Note that these different types of modularity can characterize either the whole product or a subset of the device in which case different modularity types can occur concurrently in the same device. An integral design is generally a more unitary design exhibiting a tight coupling among its subsystems and structures. A modular design on the other hand is arranged as a set of connected modules where physical modules satisfy the following definition:

> <u>Physical Module</u>: A set of components that hold together in a stable configuration with no external effort required to maintain that stability (Greer, 2002).

The fundamental problem of modularity is that of partitioning the design solution. Difficulties stem from choosing the most appropriate partitioning scheme given a multitude of constraints. Prior work has generally addressed this partitioning problem from either a functional perspective or a physical viewpoint.

Due to the recognized benefits of modularity (Cutherall, 1996; Otto and Wood, 2001), considerable interest in the research community has been directed toward the development of techniques that support modular design. Zamirowski and Otto (1999) propose a technique for developing the modularity of a portfolio of products using heuristics based on functionality. Yu et al. (1998) describe a method for identifying the appropriate type of architecture (ie. fixed-unshared, modular platform, or massively customized) based on customer needs. Sosa et al.

(2000) provide an approach for identifying modules based on an analysis of interfaces among components. Stone (1997) presents three heuristics for identifying modules based on properties of a functional model.

Both the type and the extent of modularity in a product are intimately related to product architecture. The architecture representation developed in this chapter provides a view of design modularity from multiple levels of product description including the function layout diagram, partition diagram, and the product family diagram. This section focuses on the utility of the function layout diagram with respect to product modularity and illustrates how this diagram provides a unique mechanism for identifying potential modules in terms of both functional and spatial properties.

The layout elements and layout intersections of the function layout diagram represent candidate physical modules. A module is shown as a boundary or interface while the layout elements and layout intersections provide a basis for defining these interfaces. One of the benefits of utilizing the layout elements and intersections is that each of these spatial regions is distinct in terms of functionality. It is therefore reasonable to utilize the boundaries defined by the layout elements and intersections as an indicator for partitioning physical modules. This allows a designer to identify physical modules based on functionally distinct spatial regions. The capability of separating physical regions based on function is significant because it offers a reasonably explicit scheme for reducing the functional coupling among physical modules.

Module identification is based on the designer electing to establish physical modules according to the boundaries of layout intersections and elements. An example of a function layout diagram for a Dustbuster® vacuum is given in Figure 3.10. Considering just a portion of the layout in the vicinity of the motor, several layout elements are indicated: Import hand, Actuate EE, Transmit EE, Convert EE to ME, etc. Comparing this representation with the actual device as shown in Figures 3.11 and 3.12, the architecture representation indicates some variations from the actual. The first noticeable difference is the Import Human Energy function which is manifested as the handle. The function layout diagram shows this function as a distinct module although the handle is clearly integrated into the housing shell which performs other functions like Distribute Mechanical Energy. Similarly, the Actuate Energy function is integrated into the module shown in Figure 3.12 although the representation suggests that it could be a separate module. In fact, many devices do implement such a separate physical modules as an off the shelf switch rather than an integrated unit in the Dustbuster® case.

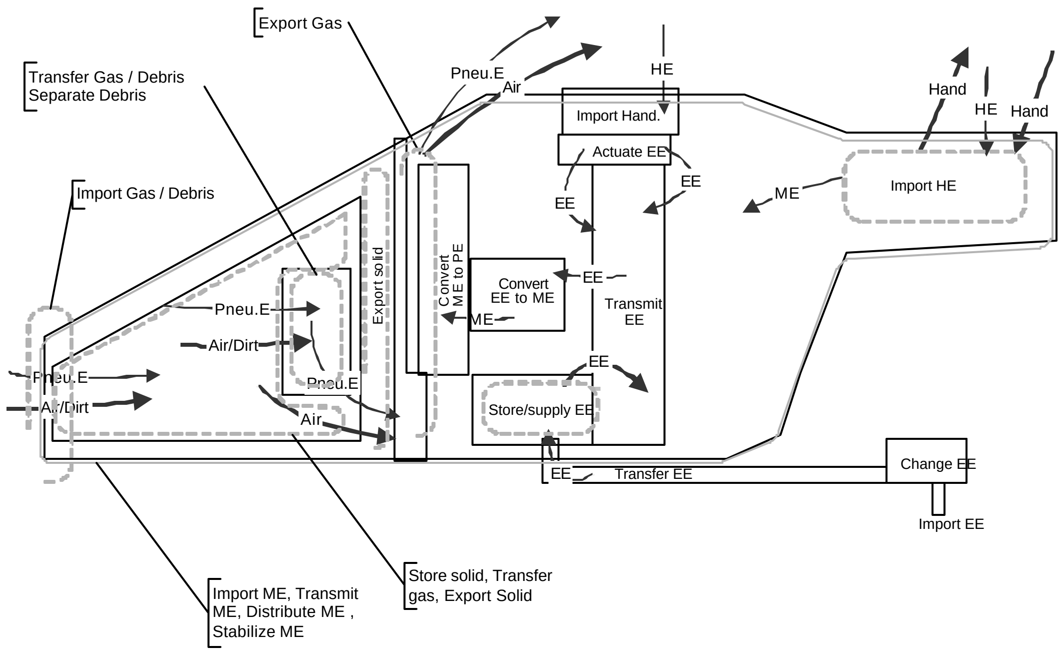


Figure 3.10 Dustbuster® Function Layout Diagram

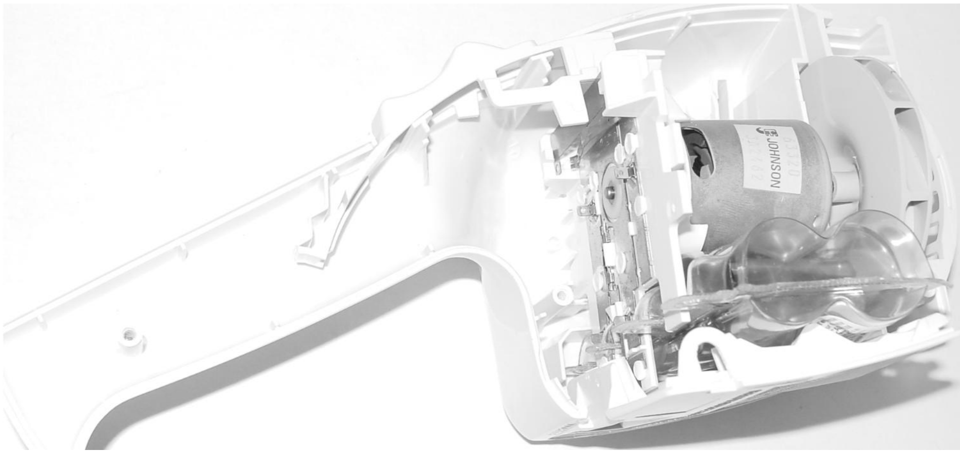


Figure 3.11 Actual Dustbuster® (front nozzle removed)

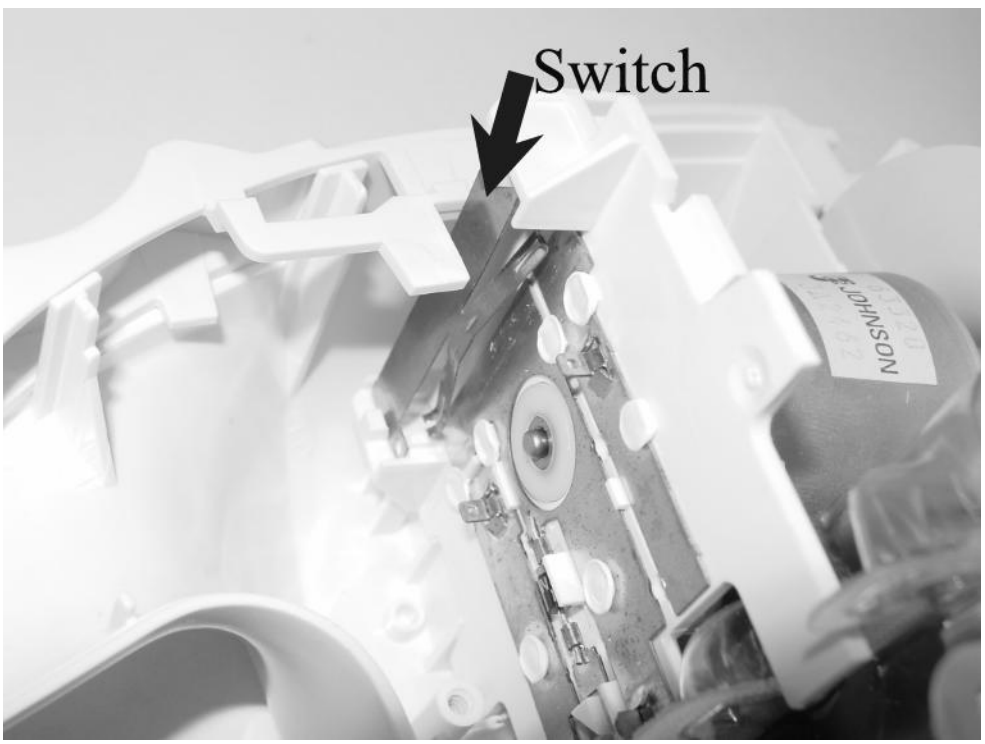


Figure 3.12 Dustbuster® switch up close

These two examples of the handle and the switch illustrate how the function layout diagram can suggest physical modules based on the spatial arrangement of functions. A brief study is performed to examine the utility of this application beyond this Dustbuster® example.

This study is performed as a part of a larger validation experiment discussed in further detail in the next chapter. Thirty consumer products are selected as a large sample set to represent the consumer product population. The function layout diagram is generated for each device and the diagram is then compared with the actual physical modules in the device. The purpose of the comparison is to determine the ratio of actual modules to potential modules indicated by the function layout diagram.

There are multiple scenarios that define the relationship between actual and predicted modules as shown in Figure 3.13. Predicted modules may contain several actual modules or an actual module may contain several predicted modules. These cases do not suggest that the function layout diagram contradicts an actual device, but instead the diagram may exhibit i) a higher or ii) lower degree of physical decomposition than the actual device, or iii) possibly different partitioning decisions but at the same level of decomposition.

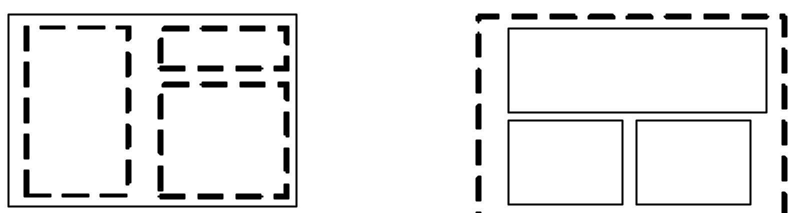


Figure 3.13 Acceptable Cases of variation between predicted and actual modules

The thirty devices involved in the validation are listed in Table 3.14 as well as the results from the comparison of actual and predicted modules. Function layout diagrams for each of the thirty products are presented in Appendix C. Branch diagrams for the same products are listed in Appendix D.

Given the overall high ratio of predicted to actual, the results indicate that further partitioning of the device is generally an option that becomes available upon inspection of the function layout diagram.

Table 3.14 Modular Guideline Validation Results

	Product	Actual	Predicted
1	GE hand blender	7	12
2	Metronome	7	10
3	Wagner paint roller	11	15
4	Skil Screwdriver	7	11
5	Freedom cordless hand sander	5	12
6	Handi work mini drill	8	14
7	Jig Saw B&D	8	18
8	Braun Coffe grinder	6	9
9	B&D Handy chopper Plus	6	12
10	Electric carving knife (Toastmaster)	7	14
11	DeWalt Palm sander	6	16
12	B&D hand mixer	7	15
13	B&D electric knife	6	14
14	GE Hand mixer	8	12
15	Bissel Hand vacuum	6	15
16	Dustbuster®	6	16
17	B&D Palm Sander	6	12
18	Handi work Screwdriver	6	12
19	Dirt Devil Spot scrubber	8	20
20	Versa Pack Saber Saw	7	12
21	Mr. Coffee ice tea maker	8	13
22	GE electric knife	7	13
23	Metro weighing scale	5	10
24	B&D cordless drill	8	13
25	3-in-one pen	7	10
26	B&D leaf blower	9	18
27	Presto salad shooter	8	13
28	Mr. Coffee maker	10	15
29	Bic Ball pen	5	14
30	Arrow light duty stapler	4	12

However, the results do not show whether such additional partitioning is a good choice since the partitioning problem is not isolated to the function layout diagram. Other difficulties compound the problem including choices of manufacturing, assembly, and the product family. Despite this complication, the predicted modules give the designer direction as to a reasonable set of potential modules should such further partitioning be needed for whatever reason. Fortunately the architecture workframe offers the designer both a set of candidate modules based on the function layout diagram as well as perspectives for other design factors that effect the end decision of where to partition the physical solution.

3.4.2 Design for Flexibility

Products generally evolve over time and at each evolution the design incurs one or more changes. Ideally, a design should require little change to evolve. It is therefore appropriate for designers to anticipate future changes and plan accordingly for those potential changes in the current design. Design for flexibility is the process of designing a product so that it is flexible to unknown future changes. There are several different definitions for flexibility and they are usually modified by some term to yield specific types of flexibility like machine flexibility, product flexibility, manufacturing flexibility, process flexibility, volume flexibility, etc. For each of the many flexibility types, there are generally multiple measures associated with quantifying the degree of flexibility. The primary issue addressed in this section is how the architecture representation supports the measurement of one type of flexibility that is typically referred to as product flexibility (Browne, 1984). The definition for this term in this paper is the following:

> <u>Definition</u>: Product Flexibility – The degree of ease with which a new product can be redesigned.

Figure 3.14 Examples of inflexible (left) and flexible (right) products

For example, consider the difference between a simple wooden chair with a more modern and adjustable model. Most would agree that if subject to redesign, the modern chair would be more flexible because there is a greater probability of part reuse. The wooden chair on the other hand is a highly integral design where most of the wood is sized and shaped such that reuse on a new generation is less likely.

This example illustrates the general concept of product flexibility and also shows that two other topics are closely related to product flexibility: design for variety and product commonality measures. A recent paper by Palani et al. (2003) develops a new product flexibility measure and briefly highlights the influence of these two related topics. This new measure is based on an analogy with Failure Modes and Effects Analysis (FMEA) techniques. The basic concept is to treat unknown design changes in a similar manner as a designer would handle potential failures in a system. The two main parameters involved are change severity and change occurrence. Severity implies the extent of influence that a design change will have throughout the product. If a particular change propagates widely throughout the device to cause other effects, then the change is more severe. Occurrence refers to the frequency which a particular change takes place. For example, external aesthetic changes for cars takes place annually.

By developing a table similar to those used in FMEA, a designer can systematically examine a product to determine the severity and occurrence for potential changes. Of course these potential changes are relevant to some local region within the device or other aspect of the device such as functionality. Several alternatives exist as potential reference frames when considering a product change. For example, one might create an FMEA style table that indexes according to function. That is, the designer would examine each and every function in the device with respect to potential changes that could occur with that function. Similarly, one might look at subsystems or components. In selecting a reasonable reference for purposes of developing a Change Modes and Effects Analysis (CMEA), the functional layout diagram of the architecture representation offers a unique feature – the inclusion of both function and form issues. Based on experiments of implementing this measure, several physical parameters are identified as affecting product flexibility. These include the following five parameters: number of functions, components, modules, standardized components, and the volume of dead space within a device.

As an extension of the above CMEA measure, a second and more objective approach is developed based on measurement of the above parameters. A comparison of the two approaches illustrates that both can yield a reasonable measure of product flexibility. However, each technique is best suited to a particular application. The first measure is very useful for the case when a designer needs an in-depth look at particular kinds of changes. This technique is relatively time intensive to use and therefore it is also most appropriate for a small set of devices. In contrast, the second measure is best suited to a larger set of devices such as a benchmark study where a designer is evaluating the flexibility for many products. This second measure is less time intensive because one can rapidly determine values for the independent parameters in the measure.

In addition to forming a useful aid in product evaluation, the architecture representation aids in designing for flexibility. First, the multi-view workframe that includes a comprehensive set of architecture information lets the designer conveniently observe product status in those areas which impact product flexibility. Secondly, guidelines are developed for directing the design toward a higher flexibility based on the knowledge of the five physical parameters discussed above. To briefly demonstrate the utility of the representation with respect to these guidelines, consider the recommendations listed in Table 3.15. The representation diagrams that utilize the factors in each guideline are also shown in the table. The capacity of the representation to accommodate observation of these factors is a direct benefit to designers interested in designing for flexibility.

Table 3.15 Abbreviated Product Flexibility Guidelines

Guideline Summary	Relevant Representation Diagram
Reduce severity by making the device modular.	Functional Layout Diagram Partition Diagram
Reduce severity by increasing the number of partitions.	Functional Layout Diagram Partition Diagram
Reduce severity by increasing the number or size of virtual or actual buffer zones.	Spatial Constraints Diagram Functional Layout Diagram
Reduce occurrence by standardizing components and interfaces.	Physical Solutions Diagram Manufacturing Diagram
Reduce occurrence by selecting technology which is far from obsolescence.	Physical Solutions Diagram

Although the guidelines above are abbreviated, a complete version is given in the next chapter.

3.5 SUMMARY

There are significant new contributions stemming from this research. The main contribution is a product architecture vocabulary and a notation in the form of a workframe. This vocabulary includes a reasonably comprehensive set of terms which provide a language for the objects of product architecture. This vocabulary simplifies the broad domain of architecture to an explicit and easily understood set of the main conceptual constituents of architecture. As a notation, the workframe establishes a new domain within the design process thus facilitating a stepping stone to effectively shorten the leap from function to form. Even within the workframe itself, the architecture design process is further partitioned into incremental steps that guide the designer from known information such as previously defined constraints to increasingly greater levels of detail and specification. Although there is still a fundamental discontinuity between function and form, the representation is a powerful aid because it directs the designer through a series of small manageable steps. A particularly attractive feature of the representation is that the workframe is very practical as it has low overhead requirements in terms of the resources needed to employ the technique. At a minimum, the representation can be implemented manually with pencil and paper. This allows for the representation to impact very quickly a potentially large audience from novice designers in academics to practicing designers in industry. In addition to the basic benefit of representing architecture, the workframe leads to new solutions for two key design activities: designing for modularity and designing for flexibility.

Chapter 4 – Guidelines

Every engineer has had the experience of witnessing both good and bad designs whether in practice or off-duty as an end user of a product. Experienced engineers have knowledge of the differences between good and bad designs that includes a range of information spanning the knowledge domain from tacit to explicit. It turns out that knowledge at both ends of this spectrum is important for problem solving and reasoning tasks (Sternberg, 1999; Boston et al., 1998). While there is active research engaged in understanding, acquiring, and deploying tacit knowledge (Sternberg, 1999), this chapter focuses on the problem of codifying tacit knowledge into an explicit form generally regarded as guidelines or heuristics. The purpose of such guidelines is to capture information that improves a designer's ability to design product architecture in a manner consistent with good design. The main chapter goals therefore are to identify good architecture designs and to document this knowledge in a set of guidelines.

4.1 INTRODUCTION AND OBJECTIVES

4.1.1 Problem Clarification

The chapter theme is closely related to the role of knowledge in design. Several questions drive the following discussion and the remainder of the chapter. What kind of design knowledge is the focus of this study? What is the knowledge deliverable of this study in terms of its function and form? How will raw data be acquired and transformed into this final deliverable? What is the impact of the deliverable and how will it be used in the future?

4.1.1.1 Guidelines as Design Knowledge

In order to codify design knowledge, it is helpful to consider the domain of knowledge that spans from tacit to explicit. Nowack (1997) presents a helpful illustration from Yoshikawa (1993) and the same concept is repeated here in Figure 4.1.

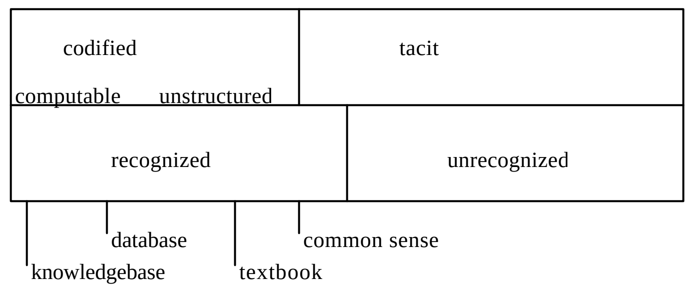


Figure 4.1 Knowledge domain.

There are significant reasons to seek codified knowledge. Compared to tacit knowledge, codified information is relatively formal, easy to document and communicate, reusable, and extensible as a foundation for future work. Codified knowledge can take many forms such as axioms, corollaries, principles, rules, physical laws, heuristics, algorithms, (Nowack, 1997) but this work focuses on guidelines. A guideline is a heuristic in that it is a recommended action that is not guaranteed to work although it generally prescribes a useful action for some condition. Originally the term heuristic meant 'to find' or 'to discover' and since then much has been written about heuristics as they apply to problem solving in general (Groner et al., 1983; Russell, 1995) and to engineering in particular (Hubka and Eder, 1996; Koen, 1971, 2003). For the work here, the following definition applies:

Definition: A guideline is a specification for some recommended action **X** on data **Y** that generally results in output **Z.**

Nowack (1997) provides a reasonable review of prior engineering design guideline work. Two related issues of prior work are significant: abstraction and guideline purpose. First, there is clearly a guideline level of abstraction that dictates the total guideline quantity and content of each guideline. Aguirre-

Esponda (1992) developed several hundred relatively low level guidelines that were organized into a hierarchy of several categories that included both product based and process based information. On the other end, Suh (1990) proposed two axioms: the independence and information axioms. Figure 4.2 below illustrates the effect of abstraction on the quantity of guidelines and Table 4.1 notes key differences between guidelines at the extreme ends of abstraction. The main point is that a more abstract set of guidelines results in a fewer guidelines.

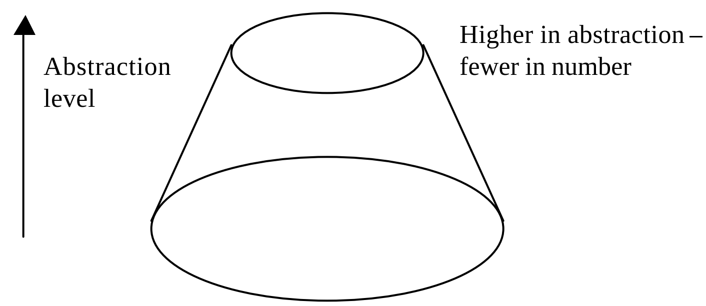


Figure 4.2 Guidelines and Abstraction Level

Table 4.1 Guideline Pros and Cons

Abstraction	**Pros**	**Cons**
High	• *Broad application* • *Few in number to maintain*	• *Difficult to implement* • *Specific cases not addressed*
Low	• *Easy to execute individually*	• *High number to maintain* • *Lack of generality*

4.1.1.2 Guideline Deliverables

Assuming that the knowledge deliverable will consist of guidelines that are at similar levels of abstraction, the problem is to define that level. Fricke (1996) presents a model of the design process that provides a basis for making this selection. He proposes a five level description of the design process and a portion of his illustration is shown in Figure 4.3. Similar partitions of the design process are given by Hubka and Eder (1996).

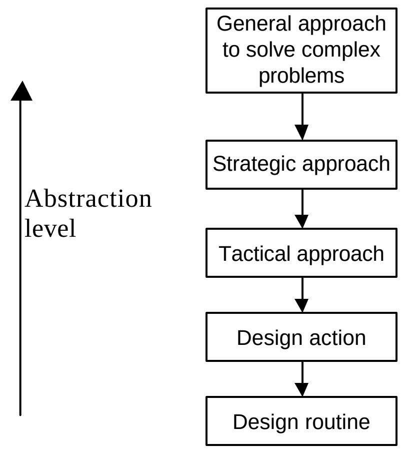


Figure 4.3 Model of Design Activity levels (Fricke, 1996)

The strategic approach above is at the same level as a design method such as the reverse engineering method (Otto and Wood, 2001) where a process is defined to direct the course of action through a sequence of steps. The architecture design method from Chapter 5 fits in this category. Tactics are mid-level actions that are narrowly focused compared to a strategy and are intended to facilitate completion of a portion of a strategy. The modeling of concept variants to determine feasibility is an example of a tactic within a design strategy. These tactics rely on lower level design actions to accomplish tactical objectives. Design actions are defined by a specific action on some specific data. An example is evaluation of stress on a part. This ‘design action’ level is consistent with the guideline definition above.

Guidelines in this chapter are developed in terms of both tactics and design actions for the following reasons. The design actions as discussed above offer an explicit template for placing design knowledge into an unambiguous and executable package. At the lowest level, these design actions define individual guideline steps. The concept of a tactic provides a useful construct for grouping closely related or dependent design actions into a meaningful unit. This allows a set of guidelines to be organized as a set of tactics where each tactic employs one or more design actions. Figure 4.4 illustrates the guideline structure adopted in this chapter.

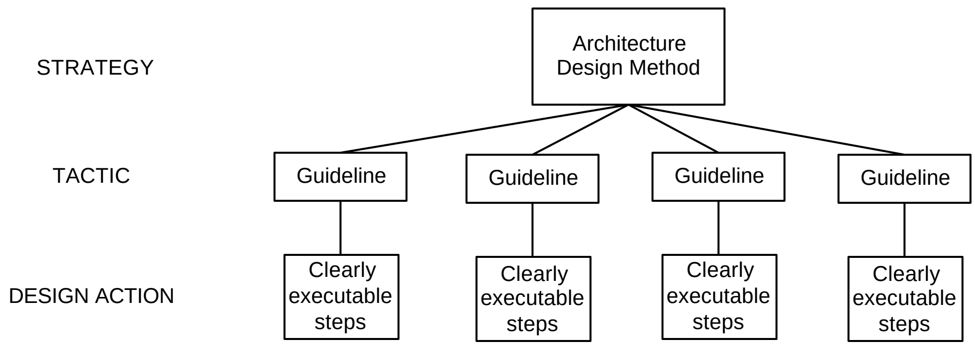


Figure 4.4 Guideline Organization

The guideline purpose is to facilitate architecture design. Because architecture design is still a relatively broad topic, there is some question about what part of architecture design is of particular interest. Later sections will discuss the approach to this question and the end result will be an approach that addresses architecture design from both a broad perspective and also with respect to particular architecture parameters.

Although guideline requirements will be presented in a later section, the basic contents of the knowledge deliverable of this work are addressed here. Nowack's (1997) work provides a foundation for this guideline content. His approach was to include at least four main parts:

1. an issue addressed,
2. reference to some design context,
3. recommended actions, and
4. supporting rationale.

4.1.1.3 Design Knowledge Procurement

Given the endpoint in terms of structure and content, how is raw data to be acquired and transformed into a meaningful set of guidelines? Generally two approaches exist: introspective and observational. In this study, the approach employs a method that begins with observation and follows with prescription (Wood and Greer, 2001; Blessing et al., 1998). The key ingredient here is the observation of empirical data. This is in contrast to an alternative approach, which is to look introspectively and prescribe the conditions of good architecture design based on the reflection of internal thoughts and experience alone. Instead, the observation and prescription cycle is a sound structure for deriving guideline results and it is the approach of this work. Specifically, the observations are made with respect to product based knowledge and an empirical study is performed to search for this data.

4.1.1.4 Guideline Application

What is the impact of the guidelines and how will they be used? The second portion of this question again addresses the process of design, which is nicely treated by Nowack (1997) in diagram that illustrates a basic action sequence. The diagram is repeated here in Figure 4.5 for clarity. As for the impact of guidelines, the intent is to provide an incremental improvement in architecture design capability with a set of novel and useful guidelines.

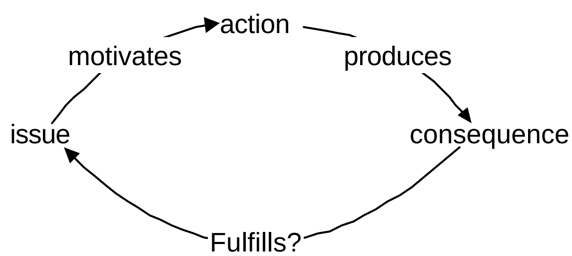


Figure 4.5 Guideline Action Sequence (Nowack, 1997)

4.1.2 Objectives

Following the somewhat lengthy clarification of the task, it is now clear that the main goal for this chapter is to search for tacit, product based knowledge of architecture design and transform this knowledge into a meaningful set of guidelines. In order to reach this goal, the objectives of this chapter are to:

1. Perform an empirical study to make observations and develop guideline content.
    a. Find a source for product based architecture design data.
    b. Make observations regarding the relation between relevant architecture parameters and corresponding design effects.
    c. Prescribe a hypothesized guideline for the observations and if possible, prescribe a causal effect based on the observations.
2. Develop a set of guidelines.
    a. Establish a format for the guidelines.
    b. Convert the information from the empirical study into guidelines.
3. Test and evaluate the validity of the guidelines with respect to a reasonable set of requirements and performance metrics.

4.1.3 Assumptions

The objectives rest on a few reasonable assumptions pertaining to the data being sought and the means of obtaining the information. First, the purpose of this chapter is to develop new, novel, and useful guidelines relevant to architecture design. These criteria are intended to exclude the redevelopment of existing product based guidelines. Secondly, the data collected in the empirical study is assumed to be a large representative of the total population of products. For this reason, the data source should have a minimum of 30 data points, which are not biased toward any one kind of device.

4.2 EMPIRICAL STUDY

The empirical study is divided into two main efforts referred to subsequently as the architecture parameter study and the architecture evolution study. First, the architecture parameter study examines relations among modularity, interfaces, parts, etc. and is performed to gain further understanding into these specific architecture parameters. The architecture evolution study is performed to examine architecture more broadly in an attempt to cover a wider swath of architecture effects.

4.2.1 Architecture Parameter Study

Data from this study was presented at the 2001 ASME DETC conference (Van Wie et al., 2001). Several questions form the motivation for investigating architecture parameters. How does the selection of interfaces and modules affect assembly cost? Is there a reasonable solution to increasing part count through careful architecture design? Do interfaces drive design? What is the relationship between architecture, cost, and complexity?

In this study, eighteen consumer products as listed in Table 4.2 below are chosen for analysis in an effort to answer the above questions. In broad strokes, the analysis involves documentation of parts, interfaces, modules, and assembly cost for each device following the teardown of the product. Assembly cost is used as the main dependent metric since this factor is relatively easy to obtain for a physical device based on Boothroyd and Dewhurst (1984) assembly time tables. Details of data collection procedure are given next.

Table 4.2 Products selected for the architecture parameter study

Mini stapler	Conair Curler	Skil Twist
Pentel Forte	Ozark Trail	Fuji
Side pencil	Revlon Curler	Conair Supermax
Swingline® small	Kodak	Remington Vortex
Swingline® large	Driving Force	B&D Drill
Coleman Quickpump	DeWalt Drill	Conair Quitetone

4.2.1.1 Data Collection Procedure

The procedure begins with disassembly during which time a list of components is compiled. The following discussion applies the procedure to a stapler. As each part is removed, the relevant information is determined in order to estimate assembly costs. The data table for the Swingline® stapler is given in Table 4.3. Handling and insertion times per part are determined using Boothroyd and Dewhurst (1984) assembly time tables. Total time per part and the final assembly costs are calculated by the same method.

Table 4.3 Example of assembly cost data collection for a stapler

QTY	Component	manual handling time per part	manual insertion time per part	Operation Time in Seconds	Assembly cost, dollars
1	base	1.13	1.5	2.63	0.011
1	staple die	1.84	5.5	7.34	0.029
1	open lever	1.13	5.5	6.63	0.027
1	die spring	1.13	10.5	11.63	0.047
1	die pin	1.88	5.5	7.38	0.03
1	flat back plate	1.69	10.5	12.19	0.049
1	btm pad	1.13	3.5	4.63	0.019
1	arm lower	1.13	1.5	2.63	0.011
1	staple guide	1.13	10	11.13	0.045
1	staple follower	1.13	1.5	2.63	0.011
1	front brace	1.13	9.5	10.63	0.043
1	upper housing	1.13	9.5	10.63	0.043
1	upper cap	1.13	5.5	6.63	0.027
1	upper spring	1.13	9.5	10.63	0.043
1	upper arm	1.13	9.5	10.63	0.043
1	spacer	1.13	9.5	10.63	0.043
1	arm pin	1.84	12	13.84	0.055
1	follower spring	1.13	10	11.13	0.045
		TOTAL		153.6	0.614

Following the teardown stage, a DSM style structure is created for each product in order to document the partitioning of assemblies and components. A consistent method for identifying assemblies is used for all products. A set of parts is considered an assembly if the set of parts could be assembled in parallel with the assembly of the rest of the product. For example, an electrical printed circuit board is an assembly to a disposable camera since the board can be put together separately from other assembly steps. The documentation of this physical structure for a stapler is given in Figure 4.7. The outer box is the whole product while each smaller box is an assembly. An assembly tree structure of the assemblies and components is derived from the DSM so that the assembly layout is clear. Two types of branches are classified and then counted: modular branches (module to module), and leaf branches (module to component).

Once the component-component layout matrix is created, the identification of interfaces is recorded by marking a “1” whenever two components possess an interface as previously defined. An inter-module interface is an interface between one or more modules and an intra-module interface does not connect one or more modules together. Since there is no implication of causal effect or direction of material or energy flow with respect to two components sharing an interface, the matrix is symmetric. As a result, only the lower triangular portion is occupied. A “1” in the shaded diagonal indicates that a part has an interface with the external environment. By external environment, this means that a part interfaces with some item other than the product itself such as the user or a desk in the case of a stapler. Representing the product in this manner allows one to fully view the layout of the product in a succinct format.

Figure 4.6 Example stapler

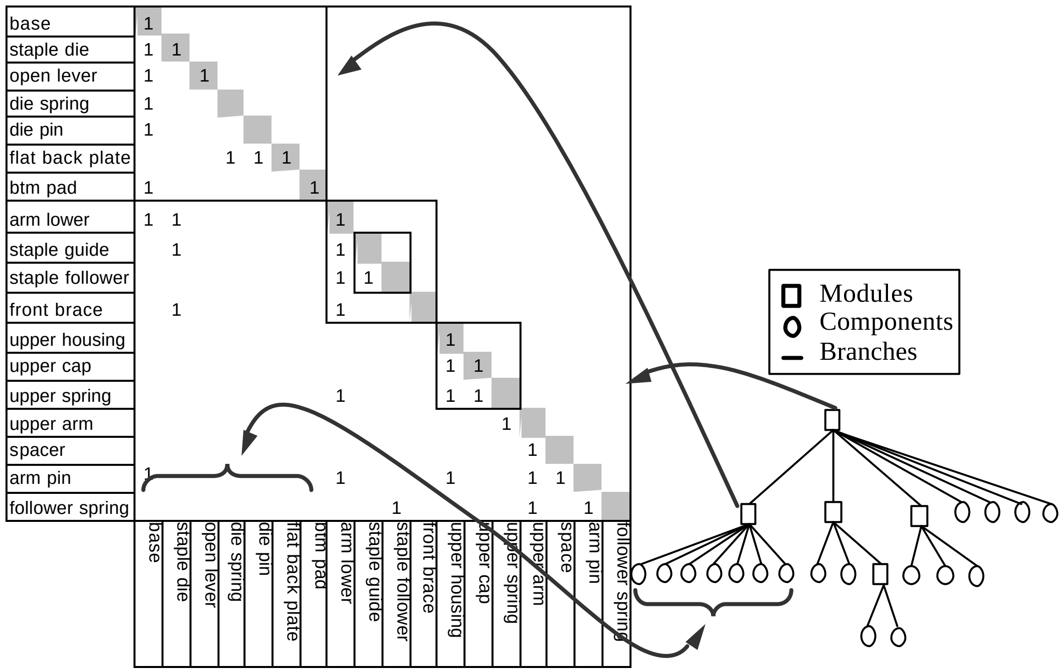


Figure 4.7 DSM and branch diagram

4.2.1.2 Results of Architecture Parameter Study

The results are summarized in Table 4.4 and arranged in a series of graphs that illustrate various relationships among relevant architecture variables. The graphs include relations among assembly cost, number of parts, number of interfaces, the type of interfaces, the type of modular layout, etc. Sensitivities for all graphs are generally within a few percent, e.g. ~5% of the overall axis range. Here sensitivity refers to the resolution of the graph in terms of a nominal change in the dependent metric given some nominal change in the independent metric. The sensitivities vary depending on the location within the graph, but the 5%

figure is taken as the average sensitivity in order show the general overall quality of the plots. In the following discussions, the references to “branches” and “interfaces” correspond to the definitions developed above. Several interpretations are taken from the graphs from the perspective of both overall trends and phenomena in local regions.

Table 4.4 Summary of results from the Architecture Parameter Study

PRODUCT	Parts	Total branches	Branches / Parts	Total Asbly cost ($)	Interfaces	Inter-module interfaces	Inter-module / Total Interfaces	Interfaces / Parts	Inter-module Interfaces / Parts	Modular branches	Leaf branches	mod brchs / parts	mod brhcs / leaf brchs	Intra-module interfaces	Ave dist from top node	Asbly cost / parts
Mini stapler	8	10	1.3	0.2	15	5	0.3	1.9	0.6	2	8	0.3	0.3	10	1.4	0.023
Pentel Forte	13	19	1.5	0.3	29	12	0.4	2.2	0.9	6	13	0.5	0.5	17	2.6	0.022
Side pencil	15	21	1.4	0.4	31	15	0.5	2.1	1	6	15	0.4	0.4	16	2.1	0.027
Swingline® small	17	21	1.2	0.5	33	13	0.4	1.9	0.8	4	17	0.2	0.2	20	1.8	0.028
Swingline® large	18	22	1.2	0.6	37	12	0.3	2.1	0.7	4	18	0.2	0.2	25	1.8	0.034
Coleman Quickpump	27	34	1.3	1	57	16	0.3	2.1	0.6	7	27	0.3	0.3	41	3	0.037
Conair Curler	34	42	1.2	1.1	62	29	0.5	1.8	0.9	8	34	0.2	0.2	33	2.5	0.032
Ozark Trail	43	52	1.2	1.3	91	26	0.3	2.1	0.6	12	43	0.3	0.3	65	3.6	0.031
Revlon Curler	43	51	1.2	1.5	68	26	0.4	1.6	0.6	8	43	0.2	0.2	42	2.8	0.035
Kodak	47	53	1.1	1.1	103	23	0.2	2.2	0.5	6	47	0.1	0.1	80	2.7	0.023
Driving Force	56	64	1.1	1.4	112	43	0.4	2	0.8	8	56	0.1	0.1	69	3.1	0.025
DeWalt Drill	56	62	1.1	1.7	134	41	0.3	2.4	0.7	8	56	0.1	0.1	93	3.2	0.03
Skil Twist	57	67	1.2	1.1	91	26	0.3	1.6	0.5	10	57	0.2	0.2	65	2.6	0.019
Fuji	58	68	1.2	1.6	156	40	0.3	2.7	0.7	10	58	0.2	0.2	116	1.1	0.027
Conair Supermax	59	71	1.2	1.8	122	59	0.5	2.1	1	11	59	0.2	0.2	63	2.9	0.03
Remington Vortex	61	70	1.1	1.7	193	79	0.4	3.2	1.3	8	62	0.1	0.1	114	3.3	0.028
B&D Drill	68	83	1.2	2	144	39	0.3	2.1	0.6	15	68	0.2	0.2	105	3.6	0.029
Conair Quitetone	69	82	1.2	2	190	94	0.5	2.8	1.4	10	69	0.1	0.1	96	2.4	0.03

Figure 4.8 shows an overall correlation between assembly cost and part count. This is consistent with the expected trend and it also follows with the generally accepted goal of reducing part count in order to reduce costs. But what about the regions where significant differences exist in assembly cost with little change in the number of total parts? This is seen in two roughly vertical areas

highlighted on the graph that contain three and six products each. In these areas, clearly something besides part count is affecting assembly cost. Apparently some types of parts are more costly to assemble than others. The basic concept in effect here is that there is a tradeoff between minimizing parts and making the parts easy to assemble. This general tradeoff is a recurring theme throughout the results.

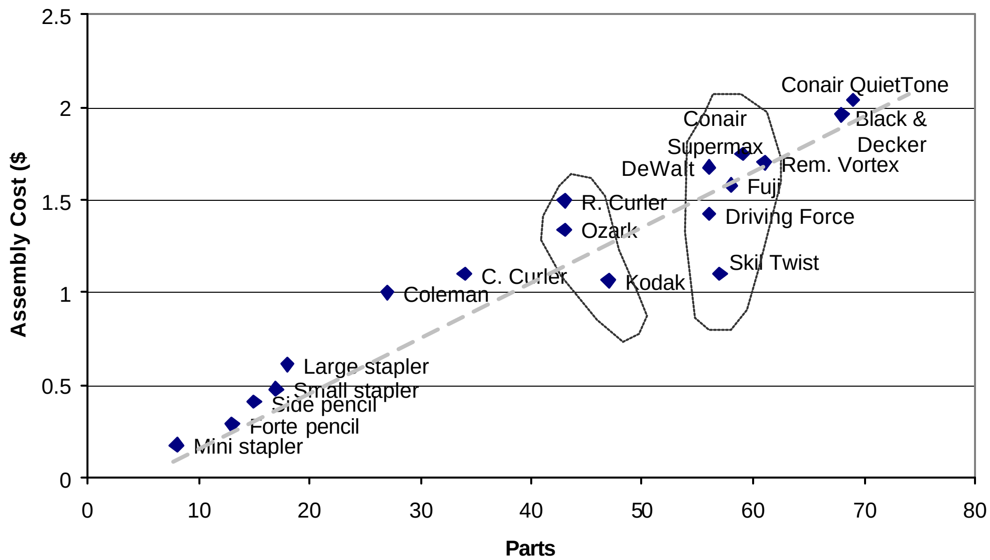


Figure 4.8 Assembly cost vs. parts

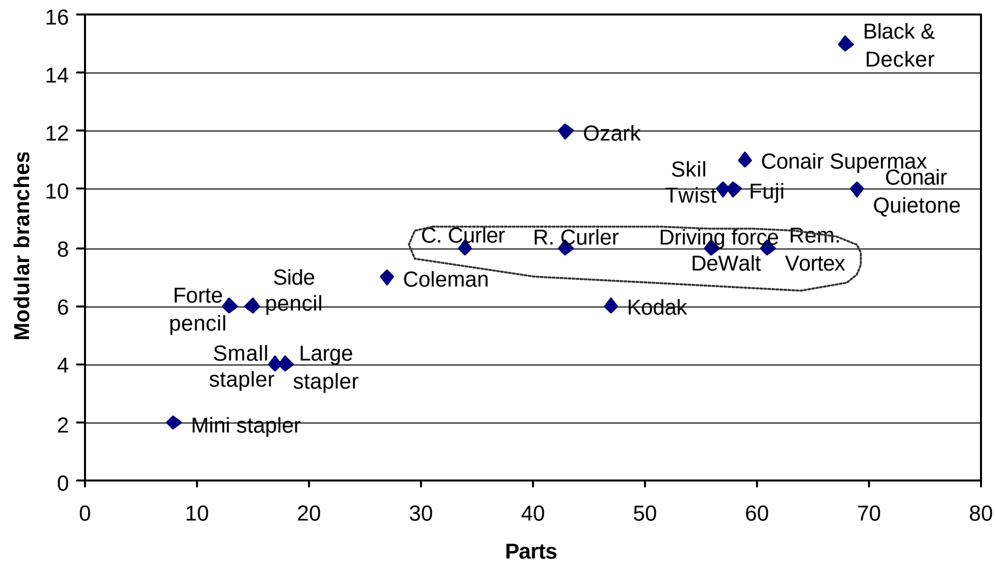


Figure 4.9 Modular branches vs. Parts

Figure 4.9 shows that as the number of parts increases, the number of modular branches, and therefore the number of modules generally increases. Implementing a modular architecture is one approach to handling increased part count, but as the graph illustrates, several products with a wide range of parts contain eight modules each. Although the general trend in Figure 4.9 presents an increased number of modules as the part count increases, Figure 4.10 shows that the ratio of the total number of branches to total number of parts decreases as the part count grows. Figure 4.11 indicates that for a given number of parts, the number of branches should be increased in order to reduce assembly cost. Similarly, Figure 4.12 indicates the same effect between assembly cost and modular branches / parts. These figures collectively suggest that effective modular design involves a consideration of the number of parts, modules, and the complexity of both.

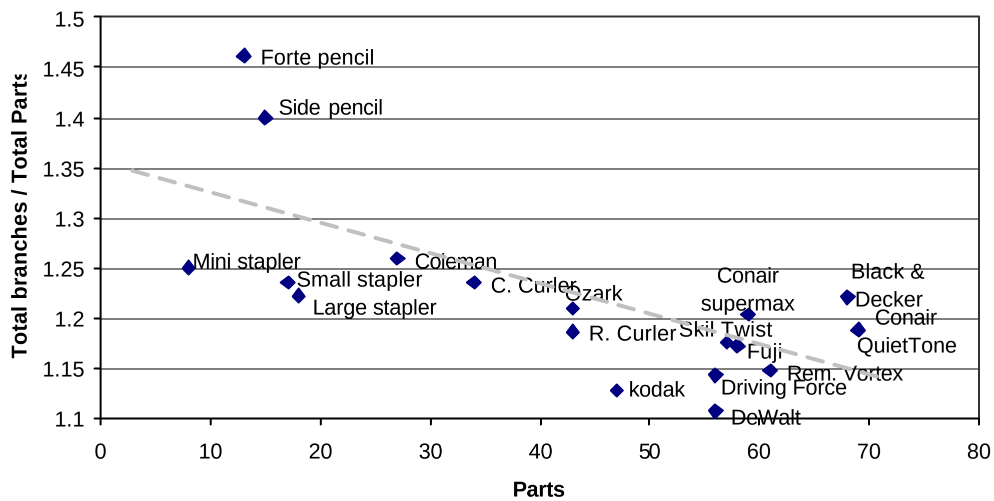


Figure 4.10 Total branches / Total Parts vs. Parts

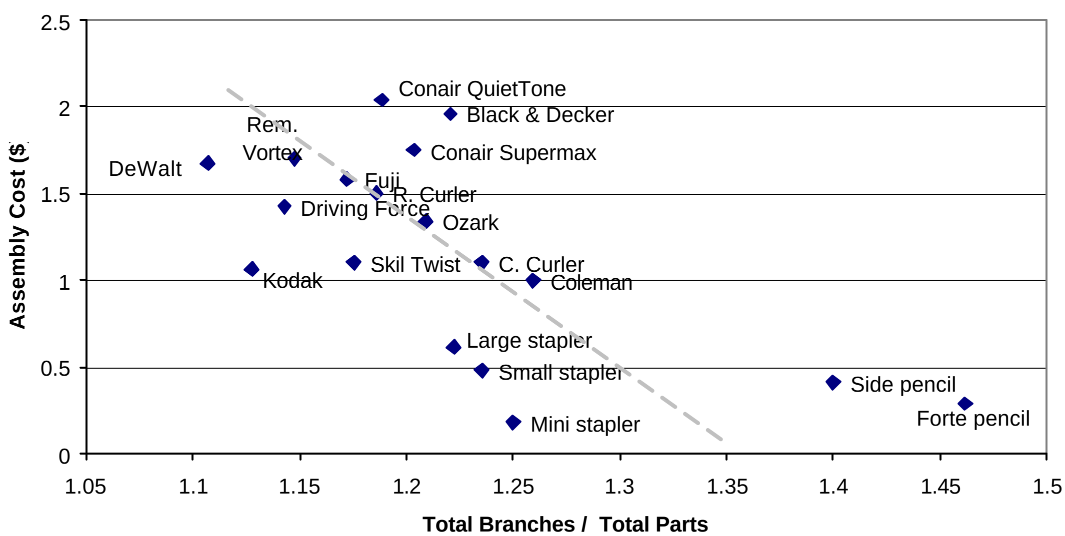


Figure 4.11 Assembly Cost vs. Total Branches / Total Parts

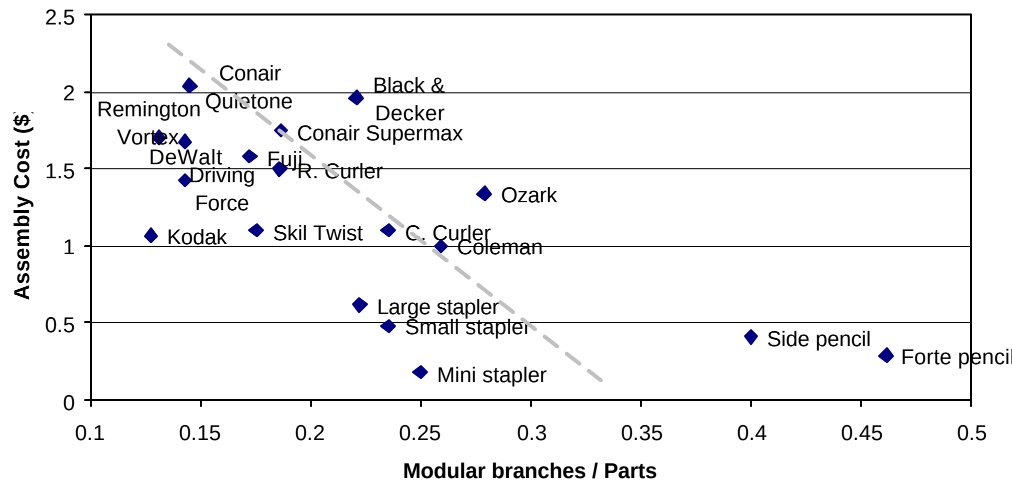


Figure 4.12 Assembly Cost vs. Modular branches / Parts

In an effort to investigate this issue, i.e., characteristics of "good" modular design, the average distance of parts from the top node of the assembly structure is evaluated. Figure 4.13 shows that the data is a sparse pattern with no clear overall trend. Given several data points in the horizontal region highlighted, this graph suggests that different layout strategies can yield very similar average assembly costs per part. By layout strategy, this implies the distribution of parts in the assembly structure. Similarly, the vertical region highlighted indicates that similar layout strategies can result in significantly different assembly costs. This is an important result because both the type of modular strategy and the manner in which the strategy is executed will affect assembly cost.

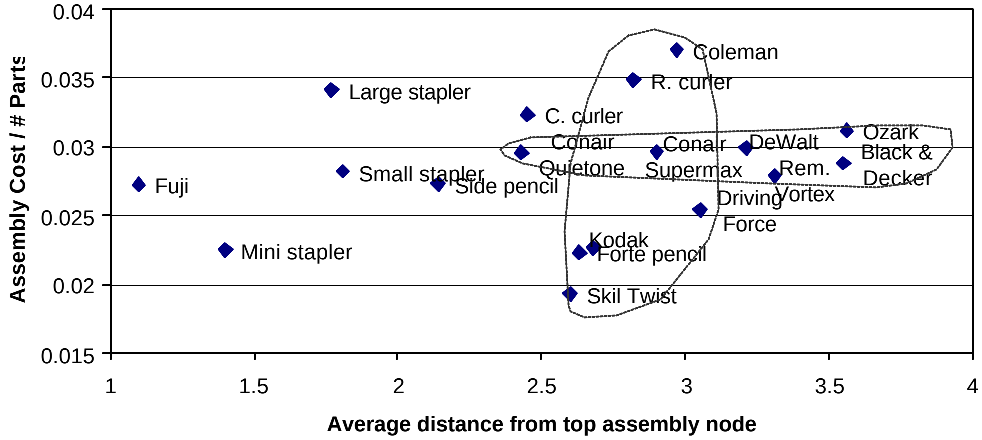


Figure 4.13 Assembly Cost vs. Number of Total Interfaces

In addition to those plots concerning the assembly structure, the relations involving interfaces are graphed. Figure 4.14 shows an expected overall trend of increasing cost associated with increased number of interfaces. Figure 4.15 illustrates another expected trend: an overall increase in number of interfaces with an increase in part count. Consider an interesting portion of Figure 4.15. What occurs in the highlighted regions in Figure 4.15 where two groups of products have significantly different numbers of interfaces, yet assembly costs do not always follow the trend of increased cost with increased number of interfaces? For example, the Remington Vortex hairdryer has many more interfaces than the Conair Supermax, but their costs are virtually identical. Similarly, the Kodak camera, Ozark air pump, and the Revlon hair curler follow the same pattern. These cases suggest that while generally an increased number of interfaces will raise assembly costs, this must be balanced with the type or complexity of interfaces being designed. The next discussion addresses two particular products from these groups in terms of complexity.

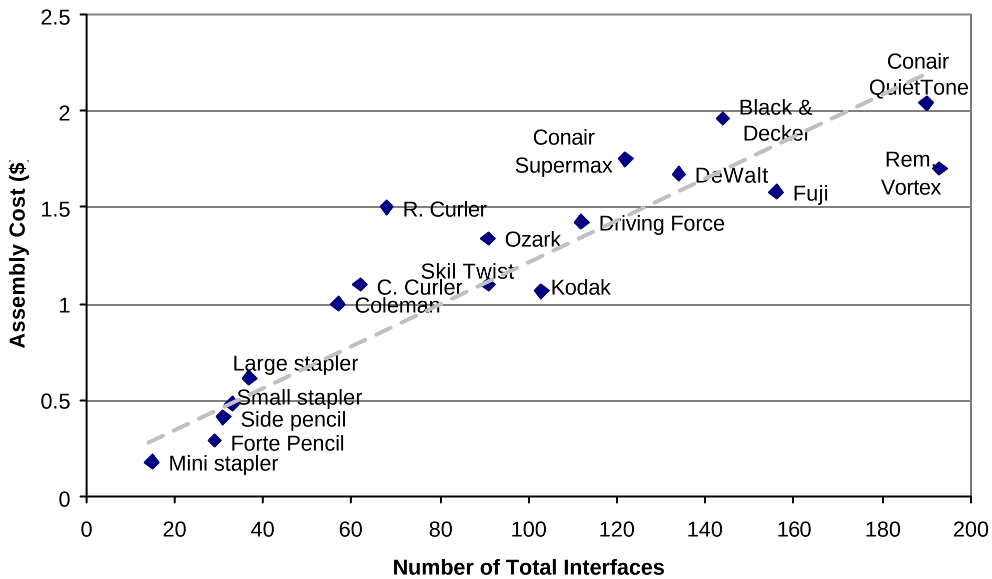


Figure 4.14   Assembly Cost vs. Number of Total Interfaces

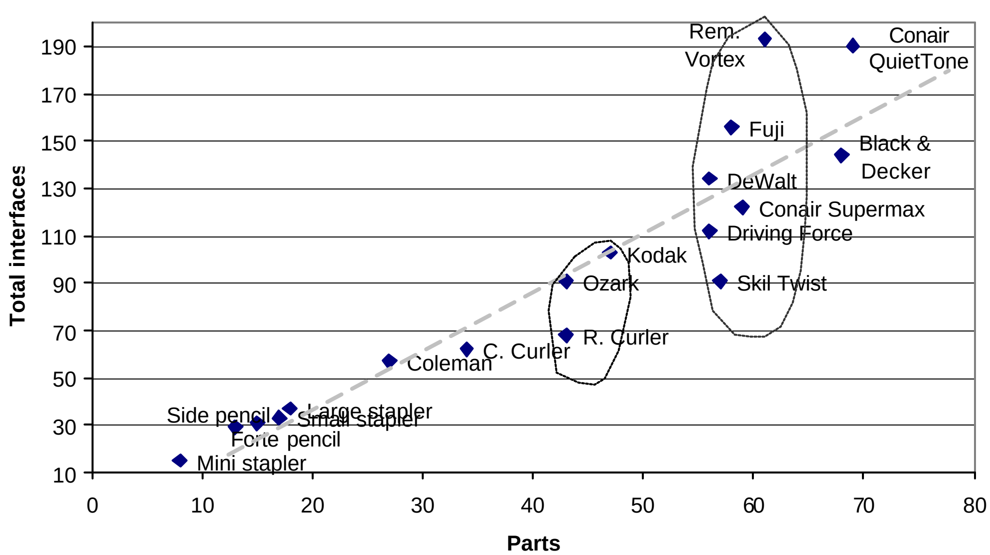


Figure 4.15   Total Interfaces vs. Parts

Notice that the Remington Vortex and the Conair Supermax hairdryers have nearly identical assembly costs. Considering that the *number of interfaces* metric is a representation of a *quantity* measure of complexity, the Remington Vortex hairdryer has a relatively high quantity measure of complexity due to the large number of interfaces. The other hairdryer with a very similar number of parts, the Conair Supermax, has far fewer interfaces. Since both hairdryers have about the same assembly cost and number of parts, the observation is that the product with greater numbers of interfaces must have less complex interfaces on average. This degree of complexity of the interfaces refers to a *difficulty* measure of complexity. The Remington model appears to have a relatively well-executed layout that involves a higher number of interfaces being assembled in about the same time as the interfaces of the Conair model. This implies that while the Remington model has a higher quantity measure of complexity, the difficulty measure is low enough compared to the Conair model that ultimately the assembly costs are about the same.

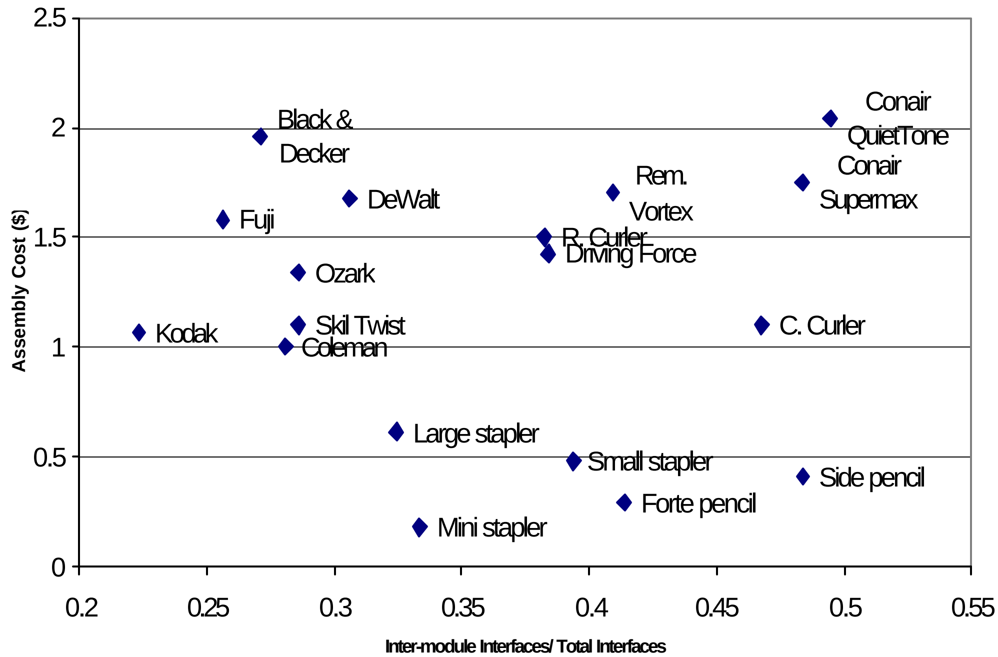


Figure 4.16 Assembly Cost vs. Inter-module interfaces / Total Interfaces

Given the effects of interfaces on assembly cost, one concern is the difference between inter-module interfaces and intra-module interfaces. Figure 4.16 shows a cloud of data with no clear trend. This implies that neither the inter-module nor intra-module interfaces affect the assembly costs more than the other. To the designer, this suggests that attention to either or both types of interfaces is important in reducing assembly costs. Figure 4.17 also illustrates the range of interfaces that can be used in order to achieve very similar values for average assembly cost per part.

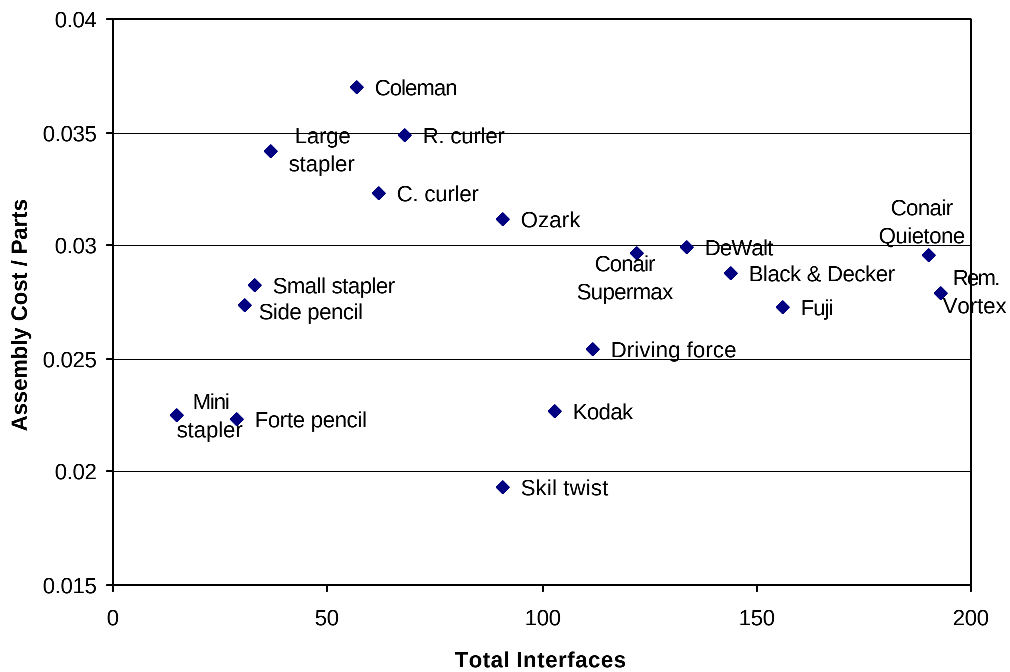


Figure 4.17 Assembly Cost / Parts vs. Total Interfaces

The trends in overall data and in local regions within some graphs support the notion that interfaces drive design by indicating the relation between interfaces and assembly cost. Given these interpretations of data, the questions that were posed earlier can now be answered:

1. How does the selection of interfaces and modules affect assembly cost?

From the data presented, the general trend is that increased numbers of interfaces will result in increased assembly cost. However, minimizing the number of interfaces must be weighed against the problem of making the interfaces too complex which can lengthen assembly times. Similarly, the use of modules can be beneficial if a suitable balance is reached between the size and number of modules.

2. *Is there a reasonable solution to increasing part count through careful architecture design?*

Yes, increased part count may be managed through careful layout in order to reduce the assembly costs normally associated with increased parts. One guideline for accomplishing this difficult task is to reduce the number of interfaces while being mindful of the tradeoff between number and complexity of interfaces. Modularization of the layout is a supplemental approach that can also be an effective strategy. Similar tradeoffs in terms of the extent of modularity and the complexity of modularity must be balanced.

3. *Do interfaces drive design?*

With the evidence from Figure 4.14, there is a clear trend showing that assembly cost follows the presence of the total number of interfaces. It is clear that indirectly or directly, interfaces affect the assembly cost and therefore drive design. Despite this relation, some designs are not given the attention toward interfaces that is needed to improve assembly costs. Additionally, this claim regarding interfaces does not dispense with the importance of other cost drivers such as material choice, final part sizing, etc.

4. *What is the relationship between architecture, cost, and complexity?*

The number of interfaces reflects both the quantity and difficulty measures of complexity. As the number of interfaces grows, assembly cost generally increases. However, there is a tradeoff between the number of interfaces and the complexity of interfaces.

From the results developed, it appears that particular aspects of architecture can be represented and evaluated to assess the value of the layout in terms of assembly cost. Developing empirical evidence of these trends impacts the design community because it allows for predictive and subsequent corrective steps to be taken during architecture design before unnecessary assembly costs are incurred. More importantly, these trends indicate that careful layout design can affect cost. This suggests that alternative layouts should be explored in order to seek an improved architecture.

The overall objective of this parameter study is to understand how interfaces in the context of architecture can be used to the designer's advantage in order to improve the quality of the product. It is shown through an evaluation of eighteen consumer products that assembly cost is affected in general by number of interfaces and that this number should be minimized as should the number of parts. The minimization however, should not come at the expense of making the parts and interfaces overly complex as this will have a negative affect on assembly cost. This tradeoff is based on the concept of reducing both the quantity and difficulty measures of complexity. Therefore, one guideline is for the designer to explore the tradeoff between minimizing the number of parts and interfaces while maintaining relatively simple levels of assembly difficulty. In addition, the data suggests that effective modular design can reduce assembly cost by allowing parallel assembly operations to take place rather than the large number of sequential assembly steps required to place the same amount of components onto a backbone one at a time. However, the number of modules should not be maximized without consideration of the complexity of both the parts and the modules.

4.2.2 Architecture Evolution Study

In contrast to the above study, which focuses narrowly on a few parameters, the evolution study extracts information on a broader set of architecture design effects. The first problem is to seek a rich and available source of tacit, product based data. Why are evolutions a good source of data? In general, products evolve according to an S-curve model as shown in Figure 4.18 (Otto and Wood, 2001). Lighting products are a good example. The transition from candles to incandescent bulbs presented a leap in performance. Similarly, the initial optimization of incandescent bulbs was a great improvement in performance in very short time. As this optimization yielded less benefits, a major shift in technology with the introduction of fluorescent bulbs initiated a new S-curve with another quick leap in performance. Of course products do not always improve, but generally the S-curve model holds true. It follows that a set of product evolutions is a good data source for this study.

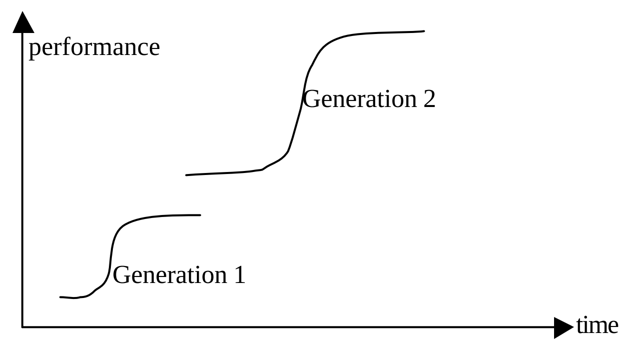


Figure 4.18 S-curve model of product evolution

For this research, evolutions for 31 devices are studied and listed in Table 4.5. Observations from these devices are given in Appendix A. Minimally each evolution consists of two devices although typically the number is greater than five and in a few cases, the number is greater than twenty. The scale of most devices is at the small and medium scale and a wide breadth of physical solutions, effects, and architectures are present in this set of evolutions.

Table 4.5 Product evolutions under investigation

Pencil sharpeners	Shaving razors	Radios
Service rifles	Ice cream spoons	Flashlights
Toasters	Pocket knives	CD holders
Staplers	Coffee makers	Umbrellas
Tractors	Hole punchers	Can openers
Writing pens and pencils	Handguns	Prosthetic legs
Chainsaws	Vacuum cleaners	Telephones
Corkscrews	Bicycles	Hair dryers
Cameras (35mm)	Key turning device	Shoes
Lathes	Reloading presses	Cars

Given the search for product evolutions, the most desirable source is a physical set of devices. The problem with this approach is that older devices are difficult to procure for study. This leads to a virtual search and two sources are useful: the internet and the literature. Those literature sources, which specifically treat the historical development of a device, are particularly useful (Hicks, 1984; Newcomb and Spurr 1989). However, these cases are more the exception. Generally, the internet is a prime source for obtaining a chronological documentation for a variety of product evolutions. A common finding is a list of pictures with a brief text description of the device. Unfortunately, the lack of physical data for these cases typically results in less than complete detail of device internals.

4.2.2.1 Observations and Hypothesized Guidelines

The purpose of this evolution based study is to gather evidence that will indicate relations between relevant architecture design variables and the quality of the design. The goal is to extract a descriptive set of observations regarding architecture design for products as they evolve. Second, the task is to associate good or evolved designs with these observations. Specific objectives overriding these two tasks include the collection of evidence that shows:

- the distinction between good and poor quality architecture design,
- this distinction in terms of architecture design variables manifested in observable device properties, and (hopefully)
- a causal effect that explains why an architecture solution results in a good design.

4.2.2.2 Data Collection Procedure

A simple and consistent three step method for making architecture observations is developed. The collection procedure is given in Table 4.6 and the full set of results is given in Appendix A.

Table 4.6 Data Collection Procedure

Step	Description
1	Find two or more devices for a product evolution.
2	Ensure that the range of devices exhibits a large degree of change over the course of the evolution.
3	Observe and document changes in the architecture as the device evolves.
4	Given a set of observations, hypothesize a guideline design action in terms of the basic guideline content as defined in section 4.1.1.2.

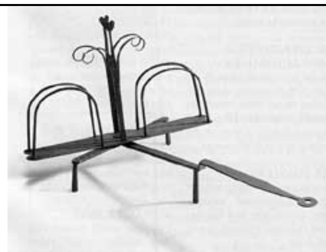
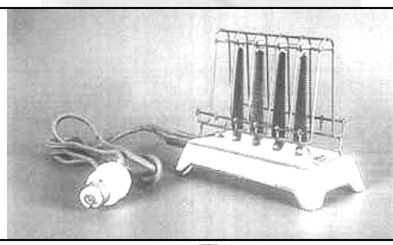
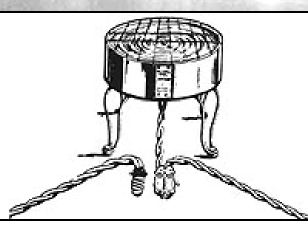
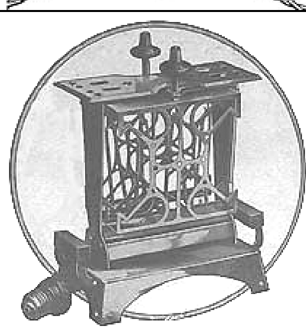
While making observations of the devices, emphasis is placed on observing the changes in parameters related to the architecture lexicon developed in Chapter 3. Typically a single observation involves multiple elements of the lexicon. The observations are documented by noting the changes that emerged during the product evolution. Following identification of one or more observations, hypothesized guidelines are developed.

Given the guideline basic content from section 4.1.1.2, generally only the action and data content was specified during generation of hypothesized guidelines. The guideline result in the case of nearly every hypothesized guideline is either a set of design alternatives or an improved or refined version of the initial data upon which the guideline was operating. The reason for leaving out the result of guideline action is that generally the consequence was implicit in the action. For example, the hypothesized guideline "minimize the length of the energy and material flowpath" states the action but does not include the result because the result is implied in the action – a reduced energy path.

4.2.2.3 Example Data Collection – Toaster Evolution

One concrete example of a product evolution is the development of the toaster. This case illustrates a typical data source and the type of observations drawn from the evolutions. Additionally in other cases, drawings, sketches, exploded views, and text descriptions were used as the raw data source. Table 4.7 shows the toaster evolution and Table 4.8 shows the hypothesized guidelines associated with that evolution.

Table 4.7 Toaster evolution (http://www.toaster.org/museumintro.html)

	Early toaster requiring a large degree of manual effort in making toast. Note that this toaster requires an external heat source.
	An early model that interfaced with a light bulb socket since at that time, standard wall outlets did not exist. This model contains a heat source.
	A stove style toaster.
	This model includes a mechanism to support loading and unloading of the bread in proximity to the heating element.

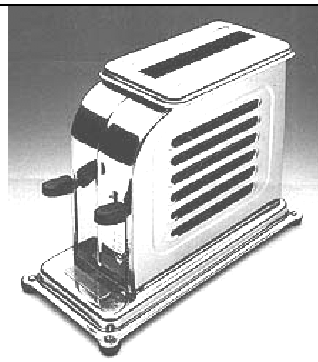
	This model includes a spring loaded mechanism for bread loading and unloading. A bread / toast storage area is located on top. Clearly, this storage feature did not persist in modern toasters.
	1919 – The architecture for the modern toaster is born. Little has changed in terms of the layout although modern toasters are most likely optimized to a greater degree.

Table 4.8 Observations and Hypothesized Guidelines

Observations: Early toasters required the user to manually turn the toast in order to heat both sides using an external heat source. Much of the layout changes were with respect to material handling issues with the toast. The pop-up toaster design in 1919, still in use today, allowed the user to place the toast into an intermediate location. The toaster would guide and store the toast in the heat area upon use of either human energy input or slow decent by the toaster in some cases. Once complete, the toaster ejects the toast into a receiving region for the user to accept.
Hypothesized Guidelines: Minimize the steps, tools, and time required for the user to operate the device. Create additional material and energy paths if access to the current operating locations is cumbersome. Reroute material and energy paths to be more accessible to the user.

Again, the full list of observations and hypothesized guidelines is given in Appendix A. The above process only develops preliminary guideline statements. This process was repeated until the accretion of preliminary guideline statements trickled toward few per evolution investigated. Given the relatively broad nature of the sample set, this trend implies that further examination of additional products could yield new guidelines but that a substantial quantity of new results would likely require the study of a disproportionately large number of devices. Furthermore, the purpose of this work is not to search exhaustively for all guidelines, but to establish a reasonable set of new ones. A plot showing the asymptotic trend is given in Figure 4.19.

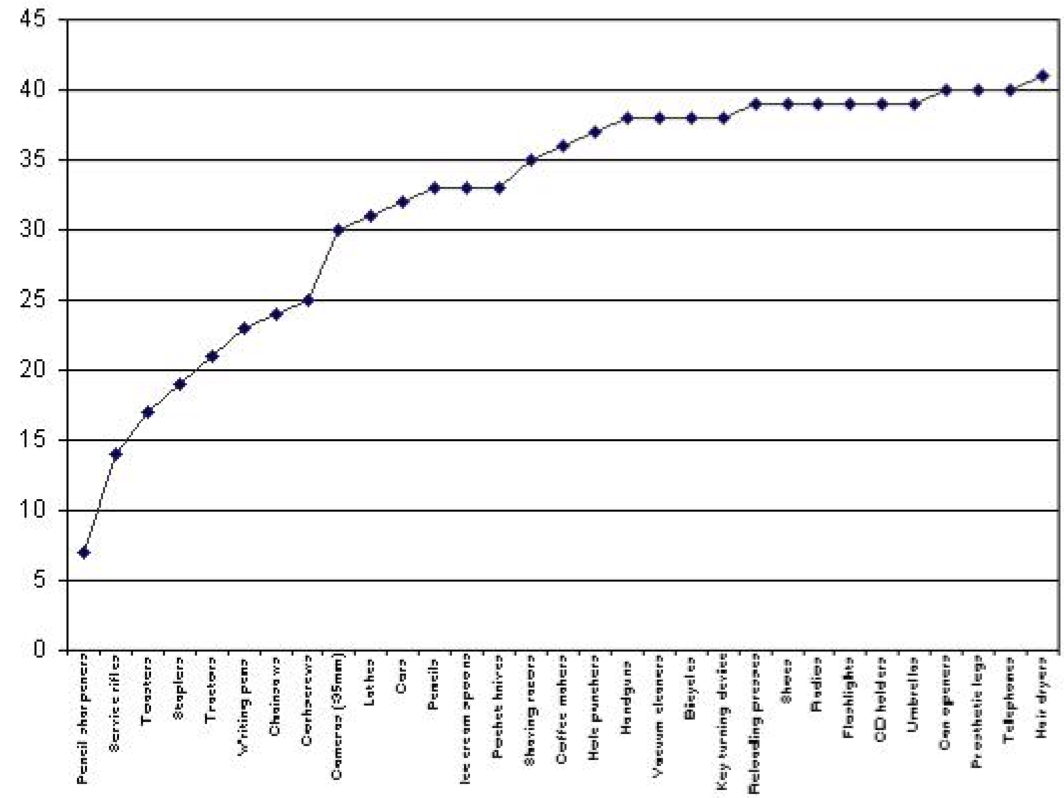


Figure 4.19 New Guidelines Generated vs. Products examined

4.3 GUIDELINE DEVELOPMENT

The previous section presents evidence that supports the execution of particular actions in order to achieve the desired result which is an improvement in architecture design. Generally the hypothesized guidelines exhibit some redundancy and lack of refinement. Additionally, the hypothesized guidelines are not in a practical format for design use because they are not organized into a meaningful set of tactics. The purpose of this section is to derive an explicit set of information from the empirical data in the form of practical guidelines. To this end, this section will accomplish three objectives:

1. Establish requirements for the final guideline form,
2. Develop a guideline format that satisfies the requirements, and
3. Refine and enumerate the guidelines according to this format.

4.3.1 Guideline Requirements

The deliverable for this section is a set of requirements that will ensure a satisfactory final guideline form. Since the total number of guidelines in this study amounts to ten in number, the logistics and organization of the guidelines is not a major problem although the format for individual guidelines is of concern. Two prior guideline development efforts provide a foundation for generating these requirements. The Federal Highway Administration sponsored a project that entailed the development of guidelines for use in the design of systems related to highway travel information, such as roadway sign design (Campbell et al., 1998). The resulting web based text from this project provides an analysis of user requirements for those using the guidelines. The analysis addressed guideline content, organization, and format with the format discussion being most relevant to this work. Table 4.9 repeats some of the top requirements based on this prior work.

Table 4.9 Guideline Format Requirements (Campbell et al., 1998)

The format should:
1. Be graphic based with supporting text.
2. Provide explicit design guidance.
3. Include text information that is brief, highly organized and tightly structured.
4. Include a rating to indicate the utility of the guidline.

Ongoing work in the medical community has produced a more comprehensive guideline format – Guideline Interchange Format (GLIF) (Ohno-Machando et al, 1998). Their model is an object oriented approach to guideline structure where each guideline consists of classes and attributes such as steps, actions, and conditional criterion. The GLIF model is well suited to computer applications where medical practitioners can access a large set of medical guidelines in a clinical environment. The GLIF model provides a structured format that includes guideline steps, action statements, criteria, and supplemental information among other things less relevant here. The following list of guideline requirements in Table 4.10 was generated upon consideration of the guideline development problem and the above prior work.

Table 4.10 Requirements for architecture design guidelines

The guideline should:
Present guideline knowledge in a consistent and concise manner.
Be independent of the method employing the guidelines.
Include graphical representations.
Provide explicit guidance to show how to apply the design knowledge.
Have succinct text where text is needed.
Include brief rational for the design knowledge.

4.3.2 Guideline Format

Based on the above requirements, the following format in Table 4.11 is adopted for the architecture guidelines. Each item in the guideline content contributes to the overall effectiveness of the format.

Table 4.11 Format Definition

Guideline Item	Specific Content
Guideline Title	An identifier which is some brief text title referring to the guideline tactic.
Guideline Recommendation	A succinct tactical level action which is some statement including the guideline action and the data on which the action is imposed.
Guideline Steps	A series of steps at the design action level that show how to execute the guideline recommendation.
Guideline Example	Graphical illustration of guideline application to a product including supporting text.

Using the above format, several guidelines for architecture design are generated from the observations and hypothesized guidelines.

4.3.3 Guideline Refinement and Enumeration

Given the hypothesized guidelines from the product evolution study, an iterative pruning and refinement of this initial data is performed in order to obtain a reasonably independent set of final guidelines. This process involves preliminary evaluation of the guidelines with respect to three primary metrics in order to determine their effectiveness. These metrics are defined in Table 4.12 and are intended to represent the issues of being practical, robust, and comprehensive. This testing was performed by issuing an assignment to students in a UT graduate design course. Their assignment was to apply the guidelines to a consumer product that was the focus of their semester-long redesign project. Upon collection of this feedback as well as ongoing evaluation on the part of the author, the guidelines are again refined to a final set that effectively addresses architecture design in reasonable manner.

Table 4.12 Guideline Evaluation Metrics

Practical, Easy to Use: This is a measure of guideline effectiveness given the designer's resources. One may also think of this measure as how much effort is required on the part of the user to execute the guideline effectively. The effectiveness is how good the guidelines help the designer perform in terms of the quality and quantity of results. Effort is a multi-faceted measure that includes the complexity of the operation, time required of the operation, amount of resources required of the operation, etc.
Technically complete and comprehensive: This is a measure the sufficiency of the guideline to comprehensively address the design issue of the guideline topic.
Consistent with Device Requirements and Customer Needs: This is a measure of how consistent the guideline's suggestions are with your device requirements and customer needs. For example, consider a guideline such as "make it lightweight." The mass or weight is relevant to both a heavy machine tool and a spacecraft, but the guideline is much more consistent with the requirements of a spacecraft since the spacecraft must be lightweight whereas the machine tool places relatively less emphasis on weight.

4.3.4 Special Guidelines – Design for Modularity and Flexibility

In addition to the guidelines developed based on observation of product evolutions, additional guidelines are developed to address two particular issues of architecture design: modularity and flexibility. These topics are chosen for study because they are both significant design considerations and the architecture representation developed in Chapter 3 leads to insight that directly facilitates development of new guidelines on these two topics.

4.3.4.1 Modularity Guideline

The previous chapter developed a foundation for using the architecture representation as a means for identifying physical modules based on functionality. Given the importance of modularity within the scope of architecture design, it is reasonable to include this design knowledge in the form of a guideline. For this reason, the partition guideline as shown in Table 4.15 includes one design action, the first action, that utilizes the module identification approach developed in the previous chapter. Designing modules for products involves two main items. The first is finding out the type of modularity and extent of modularity that the device should exhibit. Some prior work addresses this problem by developing a technique for defining the type of product architecture based on customer needs (Yu et al., 1998). The second difficulty is developing the product architecture so that the specified type and extent of modularity will exist in the product. This second issue is addressed by this new design action which is based directly on the functional layout diagram as presented in the previous chapter.

4.3.4.2 Flexibility Guideline

A flexibility guideline is also generated using a similar approach as above. Given the relation between the flexibility measure and the eight variables shown in Table 4.13, an appropriate set of guideline actions becomes apparent.

Table 4.13 Variables relevant to product flexibility measure

Number of functions
Number of parts
Number of interfaces
Level of product performance
Number of modules
Extent of virtual or actual void space
Number of standard components (OEM)
Obsolescence horizon measure

In addition to specifying the direction of change relevant to each variable, the guideline actions explain these changes in terms of change severity and change occurrence. The flexibility guideline shown in Table 4.15 presents these design actions.

4.4 GUIDELINE EVALUATION

The purpose of this section is to test the utility of the guidelines with respect to a reasonable set of performance metrics.

4.4.1 Test Criteria Development

Two primary elements of concern drive the test criteria. First is the issue of technical content. How well do the guidelines work across some expected population of products for which the guidelines should work? In this category there are two main questions that pertain guideline performance. When considering the total population of products,

1) is there evidence that a given guideline has been applied? and,
2) could the guideline be applied to the current device?

The test for this technical content issue is to evaluate a sample set of products representative of the total population and systematically evaluate each guideline with each product against the above two questions. For purposes of this test, the test questions are evaluated as true or false and recorded as 1 or 0 respectively. The standard used for the first question is whether or not a reasonable person would consider there to be reasonable evidence that suggests the guideline has been applied. The only exception to this binary assessment is the modular design action within the partition guideline. This design action is evaluated as a ratio of the actual number of modules to the predicted number of modules. The second question is tested to the standard of whether or not a reasonable person would expect to be able to impose any potential improvement in the device by applying the guideline.

The second issue that relates to guideline performance is that of communication. How well does the technical content of the guidelines transfer to the designer so that the design knowledge can be implemented in a real world application? This is largely a problem of format and the appropriate test is to compare the final guideline format with the format requirements established above. Based on inspection of the example format and consideration of the guideline requirements, it is clear that the format should effectively communicate the design knowledge to the designer.

4.4.2 Guideline Evaluation

Each guideline is evaluated with respect to the two issues discussed above. The test is performed using the same thirty devices used in the previous chapter. This sample set is not quite a random sample of products and there are some redundancies like the presence of multiple power drills. Nevertheless, the sample set is a reasonable representation of the total population of products that are mainly mechanical, at a small to medium scale in terms of size, and involve human interaction during operation. To justify the sample set, consider the variety of physical effects and functionality of the devices. Several energy domains including mechanical, electrical, pneumatic, hydraulic, and thermal are present. The devices operate in a variety of modes involving the use of solids, liquids, and gases. Many materials and manufacturing processes are represented including the use of polymers, metals, and off-the-shelf components. Some products are stand-alone while others are variants within product families. The list of devices is given in Table 3.14.

Results from the first and second questions are summarized in Table 4.14 as percentages. Each design action is evaluated with respect to the thirty devices, and the percentage shows the percentage of those products that affirmatively satisfy the two questions as shown in the table. The percentages for both questions are also appended parenthetically after each design action in final listing of the guidelines in Table 4.15. This information shown in the presentation of each guideline provides the designer with a convenient indicator of guideline utility before any attempt to use the guideline is made.

Table 4.14 Summary of guideline applicability across 30 products

Question 1(**Q1**): Is there evidence that a given guideline has been applied? Question 2(**Q2**): Could the guideline be applied to the current device?			
		Q1	**Q2**
Guideline	Guideline Steps – Design Actions	%	%
Product Boundary		100	100
	Change the size, shape, position, or orientation of the layout to make the boundary more smooth.	100	100
User Interfaces		100	100
	Alter the number, position, orientation, size, and shape of the user interfaces to be ergonomically sound and consistent with the regions of space where the user will act and the actions that the user will perform during his or her interaction with the device	100	100
Multi-Configuration		50	100
	Establish multiple product boundaries by reshaping and relocating the spatial constraints in order to accommodate each chosen activity.	50	100
Motion		100	100
	Reduce the number of motions.	100	100
	Reduce the number of changes in motions.	100	100
Partition		69	91
	Establish physical modules according to the boundaries of layout intersections and layout elements in the functional layout diagram.	52	49
	Establish internal storage modules instead of external supplies in order to improve mobility.	37	100
	Establish replacement modules or long lasting modules for regions subject to heavy wear.	100	100
	Establish modules as external attachment units.	87	100
	Establish parts within a module based on the reduction of part and interface count and their complexity.	100	100
	Reduce parts by establishing compliant mechanisms.	40	100
Structural Excess Reduction		56	83
	Reduce the overall size and mass.	90	100
	Reduce the void space (non-functioning space) within the product boundary whether the void space is either empty space or part of the device.	47	100
	Reduce the size of consumable storage compartments.	30	50
Excessive Loading		100	100
	Implement mechanical advantage to reduce excessive force.	100	100
	Use spatial tolerance, thermal effects, or lubrication to reduce frictional forces.	100	100
	Use material variations, surface treatments, or inserts to correspond to variations in stress areas of wear areas.	100	100
Energy and material flow path		83	97
	Reduce the length of the energy and material flow path.	97	100
	Reduce and regulate the flow of energy and material toward the minimum required amounts.	100	100
	Restrict energy and material flow paths to only the desired regions. (Reduce waste.)	100	100
	Include a collector or reservoir for waste or output storage.	37	87
Automation		95	100

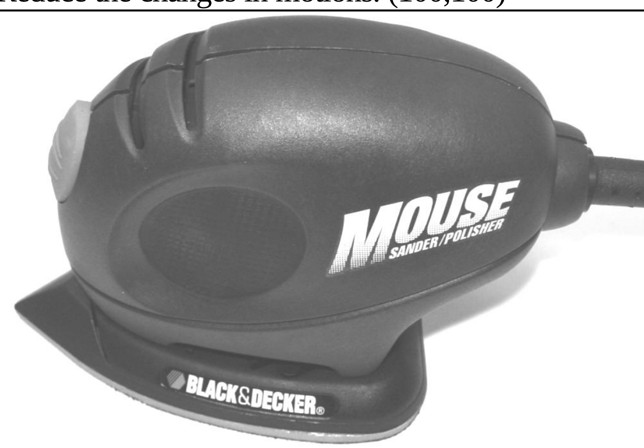
	Reduce the steps, tools, and time required for the user to operate the device.	100	100
	Replace human energy with an alternative energy source such as electrical or chemical.	90	100
Flexibility		63	100
	Reduce severity by making the device modular.	97	100
	Reduce severity by increasing the number of partitions.	3	100
	Reduce severity by increasing the number or size of virtual or actual buffer zones.	57	100
	Reduce occurance by standardizing components and interfaces.	100	100
	Reduce occurance by selecting technology which is far from obsolescence.	100	100
	Reduce severity by increasing the performance envelope of the device.	23	100
	Mean (for design actions – excludes values for the guideline overall)	77	94

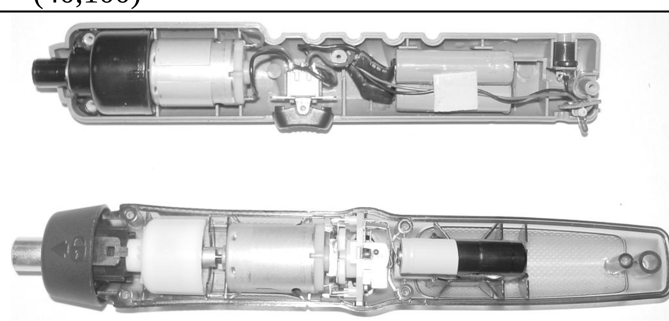
4.4.3 Guideline Evaluation Discussion

Results from the above tests show that the guidelines have generally a high rate of usage among existing products. Many guidelines are used on all devices sampled. This suggests that the guidelines are generally relevant to the existing set of products as they have evolved to their current state. The modular design action within the 'Partition' guideline showed roughly a 50% application rate among existing products so only about half of the modules identified by the guideline were actually implemented. In addition, the 'question 2' results are very encouraging because they indicate that nearly all guidelines could be used to potentially improve the sample set of devices. The final set of guidelines is given in Table 4.15.

Table 4.15 Architecture Design Guidelines

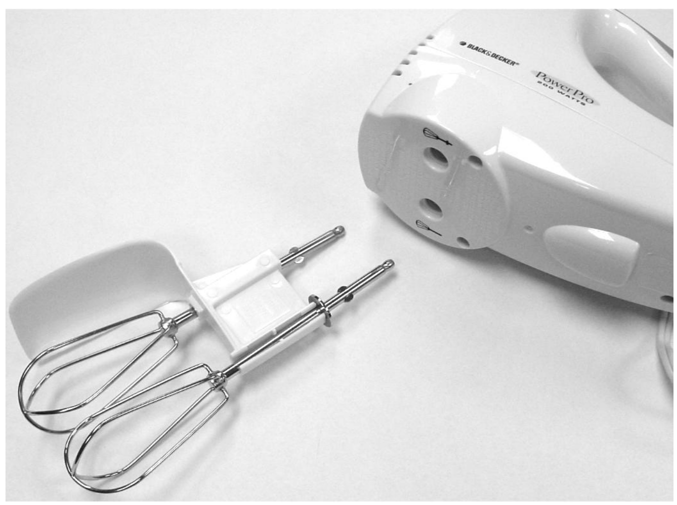
PRODUCT BOUNDARY	
Recommendation	Smooth the product boundary.
Guideline Steps	Identify areas along the boundary that protrude or make the boundary unsmooth. Change the size, shape, position, or orientation of the layout to make the boundary more smooth. (100,100)
Example	 Toastmaster Electric knife
USER INTERFACES	
Recommendation	Position user interfaces to match user activities.
Guideline Steps	Determine the actions the user will perform during use of the device. Alter the number, position, orientation, size, and shape of the user interfaces to be ergonomically sound and consistent with the regions of space where the user will act and the actions that the user will perform during his or her interaction with the device. (100,100)
Example	 Activation switch is pressed as the lid is held into place - Braun Coffee Grinder

MULTI-CONFIGURATION	
Recommendation	Generate multiple device configurations according to user activities.
Guideline Steps	Generate an activity diagram for the device. Identify activities that require different spatial layouts for the device. Identify other activities that frequently occur in parallel or in close proximity to the above activities. Establish multiple product boundaries by reshaping and relocating the spatial constraints in order to accommodate each chosen activity. (50,100)
Example	 Bissel vacuum with multiple configurations to handle multiple cleaning tasks.
MOTION	
Recommendation	Reduce device motion complexity.
Guideline Steps	Reduce the number of motions. (100,100) Reduce the changes in motions. (100,100)
Example	 This device reduces the number of motions on the part of the user.

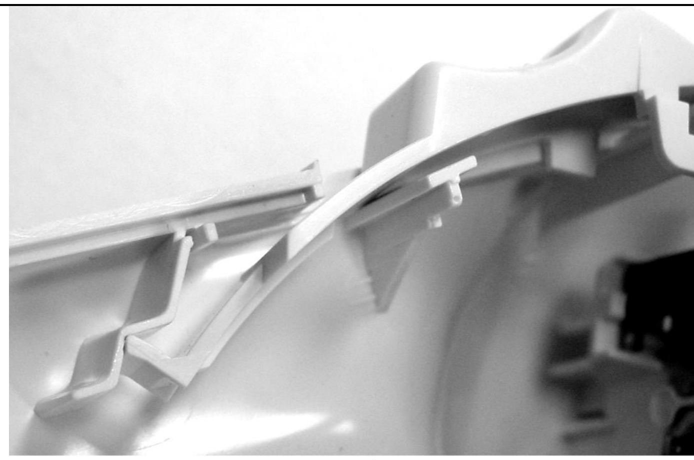
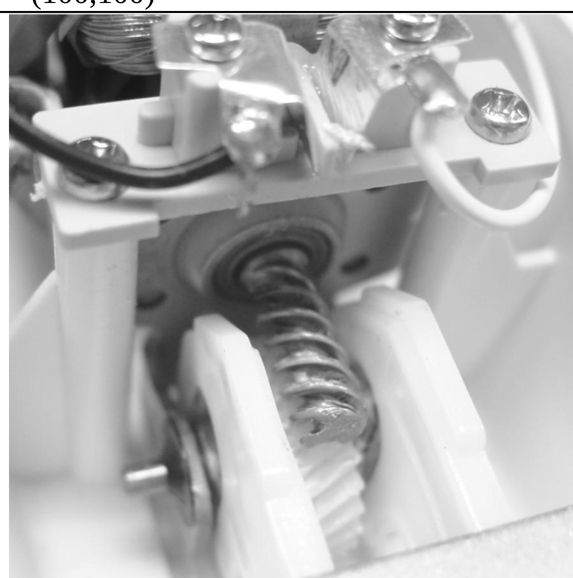
PARTITION	
Recommendation	Partition the device into appropriate modules and components.
Guideline Steps	Identify a region to be partitioned. Establish physical modules according to the boundaries of layout intersections and layout elements in the functional layout diagram. (52,100) Establish internal storage modules instead of external supplies in order to improve mobility. (37,100) Establish replacement modules or long lasting modules for regions subject to heavy wear. (100,100) Establish modules as external attachment units. (87,100) Establish parts within a module based on the reduction of part and interface count and their complexity. (100,100) Reduce parts by establishing compliant mechanisms. (40,100)
Example	 Screwdrivers – modules correlate well with the function layout diagram and in both screwdrivers, the internal storage modules (batteries) are reduced significantly in size  BIC pen – contains ink rather than old style where the user dips pen in India ink jar

DeWalt sander – uses a replaceable abrasive sheet

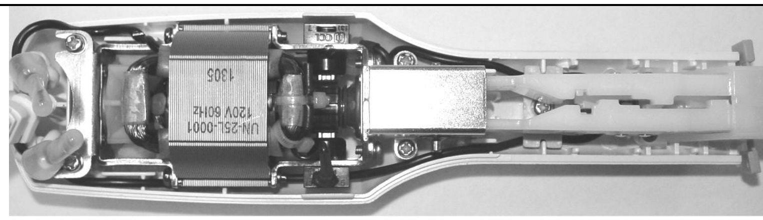
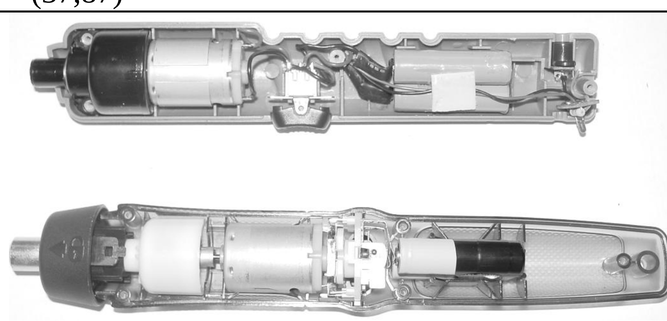
Inlet integral with the housing as opposed to the alternative below which has a separate part for the inlet.

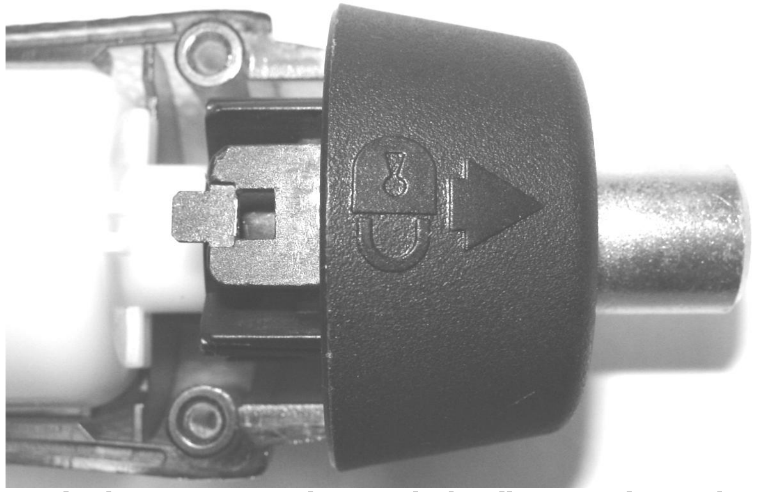
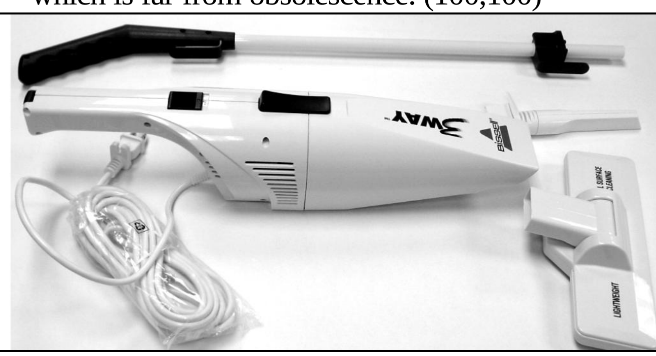


Black and Decker blender – using external attachment units

<table>
<tr><td></td><td>
Compliant switch mechanism from a Dustbuster vacuum</td></tr>
<tr><td colspan="2">EXCESSIVE LOADING</td></tr>
<tr><td>Recommendation</td><td>Reduce the negative effects of excessive loading.</td></tr>
<tr><td>Guideline Steps</td><td>Identify regions of the device where loading is excessive.
Implement mechanical advantage to reduce excessive force. (100,100)
Use spatial tolerance, thermal effects, or lubrication to reduce frictional forces. (100,100)
Use material variations, surface treatments, or inserts to correspond to variations in stress areas of wear areas. (100,100)</td></tr>
<tr><td>Example</td><td>
Increase mechanical advantage and use of lubrication to reduce friction forces – transmission in an electric knife</td></tr>
</table>

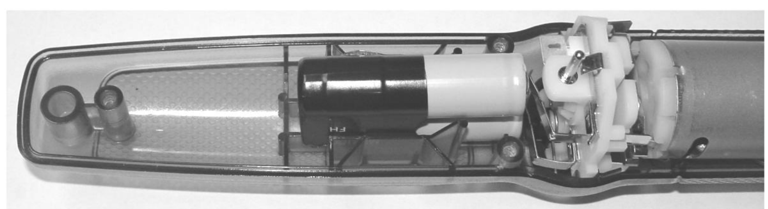
	 Food processing cutters using steel inserts
STRUCTURAL EXCESS REDUCTION	
Recommendation	Reduce unnecessary device structure.
Guideline Steps	Reduce overall size and mass. (90,100) Reduce the void space (non-functioning space) within the product boundary whether the void space is either empty space or part of the device. (47,100) Reduce the size of consumable storage compartments. (30,50)
Example	 Very small drill with a small consumable (battery)

	 Electric knife with very little void space
ENERGY AND MATERIAL FLOWPATH	
Recommendation	Reduce energy and material in the device.
Guideline Steps	Reduce the length of the energy and material flow paths. (97,100) Reduce and regulate the flow of energy and material toward the minimum required amounts. (100,100) Restrict energy and material flow paths to only the desired regions. (Reduce waste.) (100,100) Include a collector or reservoir for waste or output storage. (37,87)
Example	 The lower Skil Twist model has a shorter electrical flow path compared to the wire arrangement above. Both devices exhibit low magnitudes of energy usage and both restrict the energy flow paths to particular regions although the Skil model accomplishes this restriction more effectively given the wires on the above model.
AUTOMATION	
Recommendation	Increase device automation.
Guideline Steps	Identify a user activity related to device operation. Reduce the steps, tools, and time required for the user to operate the device. (100,100) Replace human energy with an alternative energy source such as electrical or chemical. (90,100)

Example	 A clutch on a screwdriver which allow rapid switching from power driving to manual driving. This is also an example of using an alternative energy (electrical) to replace human energy in a screwdriver application.
FLEXIBILITY	
Recommendation	Increase product flexibility to unknown future changes.
Guideline Steps	Reduce severity of changes by making the device more modular. (97,100) Reduce severity of changes by increasing the number of partitions. (3,100) Reduce severity of changes by increasing the number or size of virtual or actual buffer zones. (57,100) Reduce severity of changes by increasing the performance envelope of the device. (23,100) Reduce occurrence of changes by standardizing components and interfaces. (100,100) Reduce occurrence of changes by selecting technology which is far from obsolescence. (100,100)
Example	

The Bissel vacuum system uses multiple external modules.

The Bissel system uses a separate part for the inlet.



The Skil twist screwdriver exhibits a large amount of void space in the battery compartment. Additionally, this device demonstrates the use of a standard OEM parts (NiCd batteries and the motor) which also happen to be technology that is probably far from obsolescence.

This pen has a higher performance envelop than a typical pen due to the three modes of writing – black ink, red ink, and pencil.

These guidelines are a small subset of the whole set of general architecture guidelines. However, given the asymptotic trend in accruing further guidelines using the evolution data, it appears that the resulting guidelines are reasonably exhaustive for the particular search strategy used. The search strategy for the above guidelines includes an initial study of relations among architecture parameters followed by a product evolution study that captures design knowledge on a variety of architecture issues. In addition to the majority of guidelines, which are based on the product evolution study, the modularity and flexibility guidelines are derived from an examination of the architecture representation developed in the previous chapter.

4.5 SUMMARY

The guidelines in this chapter are a new contribution because they are a codified set of product based design knowledge that is relevant to architecture design. One of the shortcomings of prior work is the lack of focus on those product features which distinguish good architecture design from a less satisfactory case. This work gives explicit information about what design variables should be manipulated and the direction of those changes in order to produce a better architecture design. The guidelines do not apply to all cases, but the guidelines do include an indicator based on an evaluation of a large sample set that provides some clue about how useful the guidelines should be in general. In addition to several guidelines based on product evolution studies, two new guidelines for modularity and flexibility are presented.

Chapter 5 – Method

A design method is a strategy where design tasks and design deliverables are incorporated into some overall plan. A design method typically prescribes a set of specific tasks to achieve some larger goal and these strategies come in a variety. Some address the broad spectrum of design while others focus on a narrow aspect of design. Previous chapters pose the architecture design phase in a relatively narrow field of view to treat representation and guidelines. This chapter widens the perspective to address the series of actions that define how the representations and guidelines can be used in the overall design process. The purpose of this chapter is to develop and present a method for architecture design.

5.1 INTRODUCTION

A few questions guide the search for a new method. What exactly constitutes a method? What are method requirements? How should this architecture design method interface in the overall design process? How can this method be validated?

The premise for this chapter is that a new architecture design method is needed and the hypothesis is that a method based on 1) a formal representation of product architecture and 2) a set of guidelines can lead the designer to architecture design solutions in a relatively direct manner. In order to support this hypothesis, the representation and guidelines developed in previous chapters will serve as a foundation for this method. The following objectives are created to form the structure of this chapter:

1. Develop an understanding of the functions and operations of a design method.
2. Develop a set of requirements for the method.
3. Embody the method in the context of the overall design process.
4. Validate the method to confirm acceptable capability.

The result of meeting these objectives is a comprehensive specification for an architecture design method within the context of the overall design process.

5.2 TASK CLARIFICATION

The first problem to address according to the approach defined in Figure 2.1 and the first objective stated above is to develop needs based on an analysis of what a method does with respect to its users. In order to help understand the context of design method use, several items will be assessed individually including the composition, operation, and function of a method in terms of the overall design process.

5.2.1 Method Composition

A dictionary definition for a method is a way, means, or manner of proceeding (Mish, 1989). From a systems engineering viewpoint, a method consists of a set of transformation elements where each is associated with an input and output. These items are arranged in a manner that indicates sequential and or parallel paths of action (Wood and Greer, 2001). Figure 5.1 presents a simplified derivative of a Pahl and Beitz (1996) design process in which the overall task of design is shown as a flowchart. This type of format is typical for a variety of design methods in the literature.

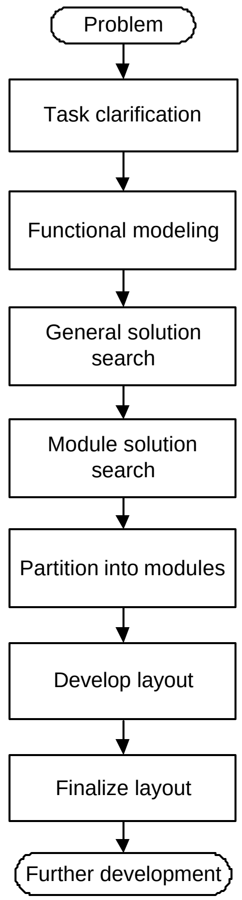


Figure 5.1 Overall design process from Pahl and Beitz (1996)

In this broad context, a method is a sequence of operations on some data. In a discussion of a systematic approach, Pahl and Beitz (1996) point out three items found in such an approach: the ability to abstract (to represent), to think logically (to know what should be done), and to think creatively (to facilitate innovation). Both the representation and guidelines developed in previous chapters are therefore major elements of the method. The architecture representation framework developed in Chapter 3 provides an abstraction in the form of a design lexicon and a design notation. This representation forms an architecture language and provides the ability to abstract architecture design through a notation that acts as a proxy for the actual design. The set of architecture guidelines from Chapter 4 provide an instruction set for thinking logically about product based data relevant to architecture design. While these guidelines direct tactical level actions as described in Chapter 4, the method offers higher level direction that can be considered strategic information. There currently exist several dozen creative techniques or strategies (Zusman and Zlotin, 1999) that can be applied to the architecture design problem. When viewed as elements of a systematic method, these facilitate product evolution by some design agent, which is a human designer in the case of this research. The overall method concept is illustrated in Figure 5.2.

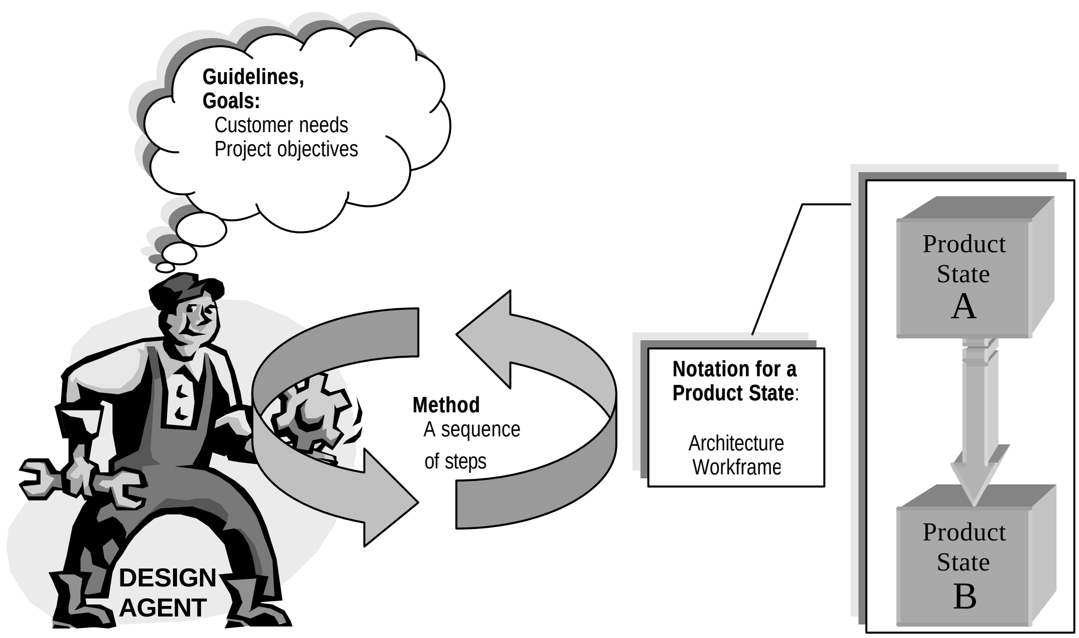


Figure 5.2 Major contributors to the architecture method

5.2.2 Method Operation and Function

In order to develop requirements, method use and function will be explored from both project scale (size) and application (original and redesign). Prior work by (Shenhar, 1998) developed a scale for the range of design projects from simple to very large and complex. Table 5.1 adopts and modifies this scale to be defined with respect to the type of design resource available. That is, a design resource can potentially be a single designer or a large firm.

Table 5.1 Design Scale

Level 1	Student design team, small project team within a design firm
Level 2	Product development group responsible for several project design teams
Level 3	Large system design – multi-year projects, eg. transportation systems

In principle, all three levels of design scales are applicable with the method from this chapter although the development has focused generally on the lower levels. Extending the method explicitly for large scale projects is left to future work. Given the targeted scale of method application, two groups of end users are expected: academic and industry users.

Academic use is aimed at both undergraduate and graduate level student use. In an educational setting, an architecture design method would likely be used by individuals and teams for both original and redesign projects. At the low end, total duration of architecture method execution would be expected to fit within a one week assignment or in the case of an expedited or partial approach, to fit within an exam period. Industrial use of an architecture design method would parallel the academic scenario with the likely exception of greater durations due to a more exhaustive search which is not artificially constrained by academic calendars.

In terms of function, the method should serve much like the synthesis process within the overall design from Figure 5.1 above. Initial inputs to this synthesis method consist of customer needs or requirements information and functional concept information. Method output is a set of alternative product architecture design solutions. With respect to this main function of generating architecture solutions, the method should exhibit improved capability compared to conventional practice. The improvement should be exhibited through a set of metrics that reflect the capacity of the method to perform well.

5.2.3 Method Requirements

Given an understanding of how the method will be used, a set of requirements is developed here to establish minimum performance standards. Table 5.2 presents these requirements.

Table 5.2 Architecture Design Method Requirements

Practical, Easy to Use	This is a measure of guideline effectiveness given the designer's resources. One may also think of this measure as how much effort is required on the part of the user to execute the guideline effectively. The effectiveness is how good the guidelines help the designer perform in terms of the quality and quantity of results. The representation must be practical to implement. This requirement is not to govern the theoretical potential limits of the representation, but to merely require that the representation be executable in some form here and today.
Complete and Comprehensive	This is a measure of the sufficiency of the guideline to comprehensively address the design issue that the guideline is intended to address.
Interface well with the overall design process	The method must have acceptable inputs and outputs that interface with the design process before and after architecture design.
Support generation of alternative architecture concepts	The method must provide a mechanism for generating multiple alternative solutions.
Systematic	The method should be systematic in the sense that a designer has clear and explicit steps to follow – a strategy.
Be robust to typical design project noise	The method must work despite reasonable variations in designer skill, product domain, project type (including design process standards imposed by contractual agreements such as defense contract projects), and designer resources.

5.3 METHOD DEVELOPMENT

The method is developed in a top-down approach beginning with its placement in the design process. The task of this development is to create a cohesive process based on the main architecture elements given above in Figure 5.2. Following development of the method framework, method execution is discussed in terms of the steps performed during architecture design.

5.3.1 Method Framework

Architecture design occurs as a synthesis process much like that shown in Figure 5.1. The architecture method is shown below in Figure 5.3 with respect to the remainder of the design process. The following sections examine the method in increasing levels of detail until the method is completely specified.

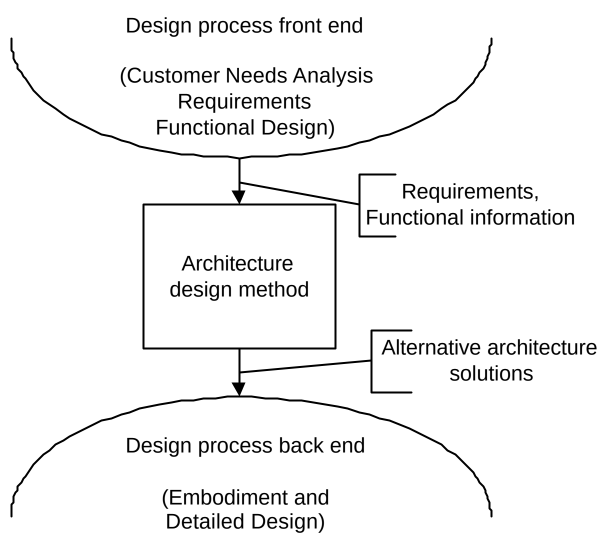


Figure 5.3 Architecture design with respect to the design process.

5.3.1.1 Method Workframe

The architecture design process is manifested through a representation of the actual design artifact. This representation is illustrated below in Figure 5.4

with a shaded outline. The representation provides elements used during operations of generation, observation, manipulation, evaluation, etc.

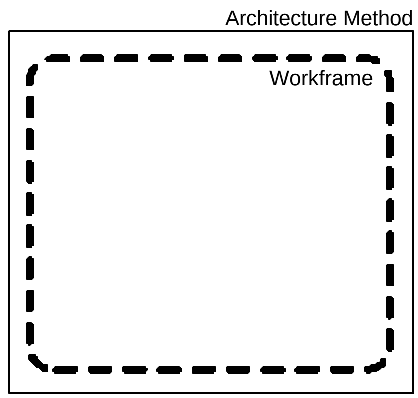


Figure 5.4 Architecture workframe

This workframe consists of the six representation elements developed in Chapter 3. These six items are shown below in Figure 5.5.

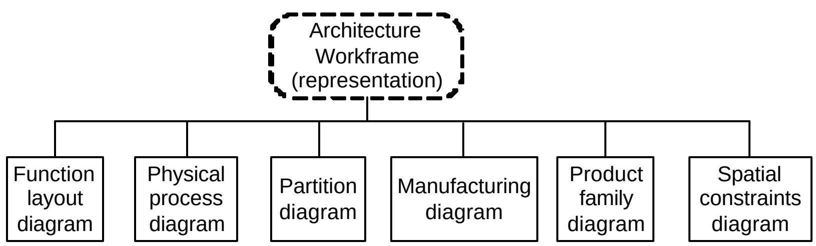


Figure 5.5 Architecture design representation details

5.3.1.2 Design Action Cycle

The design action cycle is a low level process that defines the set of actions a designer can perform in order to manipulate the design. At this basic operational level, the representation is manipulated in terms of additions, deletions, or changes. That is, a designer can add items to the architecture workframe such as additional modules, partitions, candidate physical solutions, or any other artifact represented in the workframe. Similarly a designer may delete or change these items. Different models of these iterative design operations have been proposed (Campbell, 1998; Nowack, 1997) and Nowack's version is adopted here as the low level design action cycle. This action sequence is shown in Figure 5.6 and indicates the general process used to observe and control the representation.

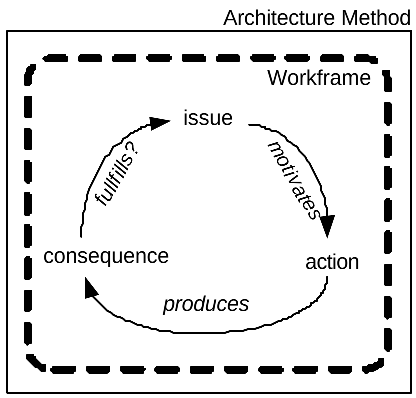


Figure 5.6 Design Action Cycle (Nowack, 1997)

Thus far, the method specifies a representation that facilitates observation and manipulation of the design. What is now needed is a mechanism for providing design direction with respect to performing the design action cycle in a manner that fosters innovation and is consistent with good architecture design practice.

5.3.1.3 Innovation Process Module

Much of design is refinement, modification, and evolution of some set of concepts under consideration. At some point necessarily, the concepts come into being whether as an original idea or as a legacy artifact from a previous system. Additionally, design fluctuates between converging and diverging activities. (Roozenburg and Eekels, 1995). These processes directly support the generation of multiple alternative solutions. For this reason, an innovation process module is included in the method as a means to provide knowledge about generating architecture design ideas. This module, shown in Figure 5.7, is a set of creative techniques that individually offer a specific approach for generating new ideas for a particular problem. Collectively, they are a rich source of different techniques that is useful when new ideas and concepts are needed.

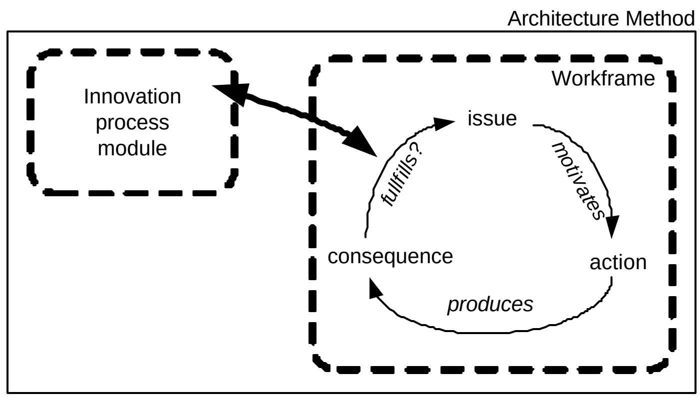


Figure 5.7 Innovation process module

The specific contents for the innovation process module are based on a variety of previously developed creative techniques. As a source for browsing and selecting candidate creative techniques, Zusman and Zlotin (1999) offer a summary of over ninety different creative methods such as mind mapping and brainstorming. Some of the techniques are referenced and well known, others are referenced informally with the name of the source institution only for example, and a few are a bit dubious such as the "Unconscious Problem Solving" method that does not include a reference. For purposes of this research, only those techniques that are reasonably well founded are included here. These techniques, which represent the constituents of the innovation process module, are listed in Table 5.3.

Table 5.3 Creative techniques for idea generation

Technique	Source
Brainstorming	Osborne
Forced analogy	McAdams
Morphological Analysis	Zwicky
Mind Maps	Tony Buzan
6-3-5	Otto and Wood

5.3.1.4 Guideline Module

Guidelines as developed in Chapter 4 are an appropriate resource for design knowledge that indicates how the designer should handle particular issues in order to develop a good architecture design. Like the previous module, the guideline module consists of design guidance at the tactical level. Figure 5.8 illustrates the method with this module included and interfacing with the design action cycle.

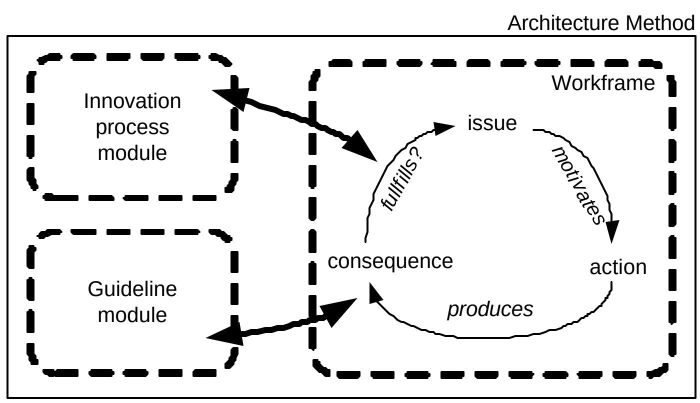


Figure 5.8 Method with Guideline Module included.

The guideline module consists of the same set of guidelines developed in Chapter 4. Table 5.4. summarizes this set.

Table 5.4 Guidelines in the Guideline Module

Product Boundary	Excessive Loading
User Interfaces	Structural Excess Reduction
Multi-Configuration	Energy and Material Flowpath
Motion	Automation
Partition	Flexibility

5.3.2 Method Execution

So far, the relation among method elements has been established without much reference to the sequence of operations a designer performs in order to apply the method. The following sections discuss this sequence of operations and present details of design tasks during each step for clarification.

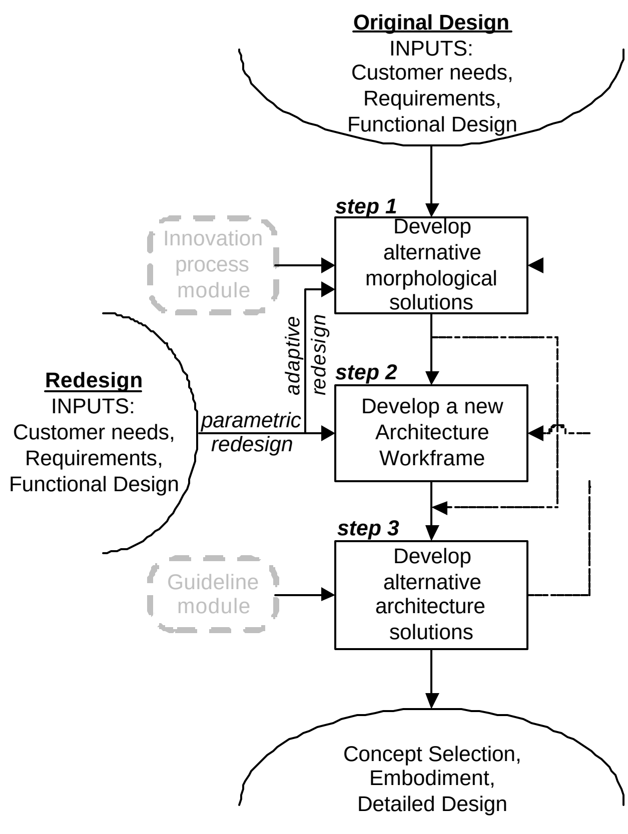


Figure 5.9 Architecture Design Method Sequence of Operations

5.3.2.1 Method Sequence of Operations

The method is executed on three levels: strategic, tactical, and design action, with two possible input ports (original design and redesign) and one output port (embodiment). Strategic execution is defined with three main steps, discussed individually in detail below, and their order of operation. The method as presented in Figure 5.9 illustrates the strategy. The dotted arrows suggest an alternative path to the solid arrows, which indicate the most probable course of action in general. Given that only three main steps and a few paths of action are defined, the strategy for architecture design is relatively simple.

5.3.2.2 Step1 – Develop Alternative Morphological Solutions

The purpose of this design step is to facilitate the generation of alternative physical solutions that satisfy some function. This step is a direct transformation of function to form. However, this step is restricted to only the task of generating morphological solutions given some function. Other baggage normally associated with conceptual design is relegated to the other two steps in the method. For example, concerns of part count, interface complexity, manufacturing technique, assembly choices, etc. are not directly addressed in the generation of morphological solutions. A designer may choose to consider any design issue at any time, but as a recommended approach, the method specifies morphological solution generation as a distinct step. This strategy is in keeping with a more general problem solving approach, which is to decompose a problem into more easily solvable portions. The specific tasks of this design step are given in Table 5.5 using a similar descriptive format as Stone (1997).

Table 5.5 Step 1 – Develop Alternative Physical Solutions

Task	Input	Output	Use / Impact
1. Identify a function or set of functions to transform into alternative solutions	Function structure Functional module Workframe (optional)	Basic function Supporting function Functional modules Set of functions not yet defined as a module	• Focus creative efforts on a particular functional aspect of the device
2. Select one or more creative tactics from the innovation process module	Innovation process module	One or more creative tactics.	• Utilize a creative tactic that suits the particular design problem at hand
3. Use the selected creative tactic to generate alternative physical solutions	Basic function Supporting function Functional modules Set of functions not yet defined as a module One or more creative tactics. Customer needs Design requirements	A set of alternative physical solutions that satisfy a set of functions (often arranged in a morphological matrix)	• Produce a variety of reasonable solutions which can later be pruned and refined

5.3.2.3 Step 2 – Develop a New Architecture Workframe

The purpose of this step is to initialize an architecture representation and consequently set up the workframe for exploring and exploiting architecture alternatives. Individual tasks and the sequence of tasks are driven in part by the information required to complete each task. Figure 5.10 shows the initiation sequence for creating a new architecture workframe. The sequence is based on the minimum upstream knowledge required to establish a given workframe diagram. For example, a product family diagram may be created based on the function layout despite the likelihood that a more detailed product family diagram could be generated if the partition diagram was already defined. In addition, the sequence implies that for a given flow, that flow includes all upstream information. For example, customer needs information is not shown explicitly as a flow into the function layout diagram although this information is implied by way of the flow from the spatial constraints diagram. The initiation sequence specifies a reasonable order of operations to create the workframe.

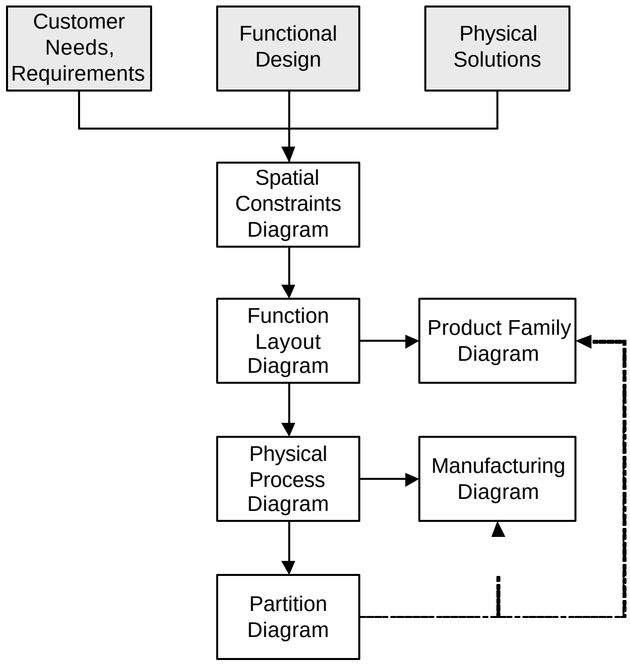


Figure 5.10 Initiation sequence for workframe creation

There are two possible entry points into this design step: original design and redesign. For the case of original design, the design minimally includes spatial information based on customer needs, requirements, and morphological solution alternatives from step 1. Alternatively a redesign entry brings much more spatial information. Depending on the type and extent of the redesign, the designer may elect to perform step 1 before step 2. Logically, a parametric redesign maintains much of the same architecture whereas an adaptive redesign incurs a greater architectural change due to functional additions or changes. For the redesign case, the initiation sequence above is less useful since one can generally begin creating the workframe with any of the workframe diagrams. Because the designer typically will have less information in the original design case, this less flexible case is considered in the following table for purposes of developing step 2 design tasks.

Table 5.6 Step 2 – Develop a new architecture workframe

Task	Input	Output	Use / Impact
1. Establish a spatial constraints diagram	Customer needs Requirements One physical solution Black box	Spatial constraints and industrial design syntactics	• Establishes boundary conditions including all external flows. • Establishes product silhouette which can be a baseline for developing industrial design syntactics
2. Establish a functional layout diagram	Functional solution A Physical solution alternative Spatial constraints	Function to form mapping Physical flows Physical interfaces Candidate physical modules	• Associates functionality with spatial properties • Identifies candidate physical modules
3. Establish a physical solution diagram	Functional design Physical solution alternatives	Physical solution choices Relative motion	• Associate alternative physical solutions with a given set of functions
4. Establish a partition diagram	Physical solution diagram	Physical partitioning	• Establishes a decomposition of the physical solution into components
5. Establish a manufacturing diagram	Physical solution diagram	Manufacturing choices	• Associate a manufacturing choice with a region of the device
6. Establish a product family diagram	Function layout diagram or a Physical solution diagram or a Partition diagram	Product family elements	• Establishes both common and distinct types of product regions.

5.3.2.4 Step 3 – Develop Alternative Architecture Solutions

This step refines and develops the preliminary information established in the workframe initiation step. The tasks for this step are based on the application of guidelines from the guideline module as given in Table 5.6. The step 3 process is to iteratively observe the workframe and evolve the workframe based on tactical knowledge from the guideline module. Table 5.7 below defines these steps.

Table 5.7 Step 3 – Develop alternative architecture solutions

Task	Input	Output	Use / Impact
1. Identify a design opportunity by observing the workframe and searching for underdeveloped or unspecified aspects	Workframe	Design opportunity	• Search for design issues open for modification
2. Identify a design guideline relevant to the design opportunity	Design opportunity Guideline module	Selected design tactic	• Procurement of relevant design guidance
3. Interpret the best course of action based on the design tactic and the specific problem	Design tactic Design problem	Selected design action	• Specific knowledge of how to evolve the design
4. Execute the design action	Workframe Design action	Altered workframe	• Low level manipulation of the architecture solution
5. Reflect and repeat task 1 until done	Altered workframe	New perspective of the design solution	• Observe the effect of changes

5.4 METHOD EXAMPLE – ORIGINAL DESIGN

In order to illustrate the utility of the method and to clarify how it applies to real world design problems, an original design is performed to design a rifle platform for use in precision mounting of a rifle during ammunition testing. The design is presented from start to finish with the main discussion focusing on architecture design and implementation of the proposed architecture method.

5.4.1 Problem Definition

The problem is to design a device that will reduce human errors that result in loss of precision when running “design of experiments” type tests to evaluate ammunition for determining the optimum load parameters for one or more rifles. Generally when a new load is being developed, the user test fires multiple batches of ammunition using a simple rest or support. This technique is sufficient for most purposes although this technique can be improved to increase the repeatability of the rifle position during a string of shots. The fundamental problem is in the users inability to return the rifle to the same position for each round fired. This error is generally caused by visual deficiencies which is aggravated in the case of an iron sighted (non-scoped) rifle, and most often the users inadvertent movement during firing. The objective of this original design is to develop a new device that reduces these human errors.

5.4.2 Pre-Architecture Design

A few assumptions are relevant here. The customer for this device is the author and the production is limited to one prototype device. Additionally, the goal is twofold: to demonstrate the architecture design method and to deliver a working prototype that will be used. A list of customer needs is given in Table 5.8 and Figure 5.11 illustrates the activities associated with device operation. A black box and function structure follow in Figures 5.12 and 5.13.

Table 5.8 Customer Needs

Lightweight (<50 lbs)
Small size (fit easily in truck bed)
Low Cost (<$100)
Usable on a rigid bench, concrete floor, and earth surface
Usable mainly for an AR-15 although should be extensible to generally all conventional (non-bullpup design) rifles
Adjustable for elevation and windage

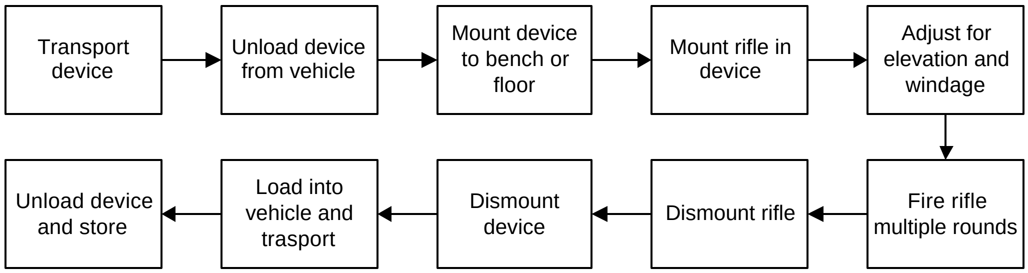


Figure 5.11 Activity Diagram for the machine rest

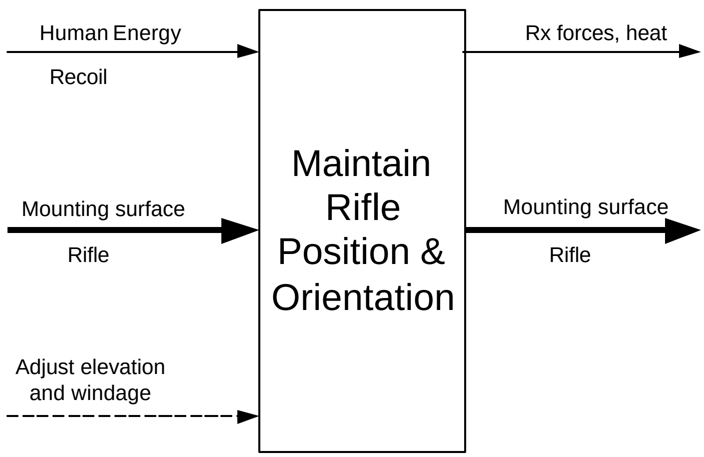


Figure 5.12 Black Box for the machine rest

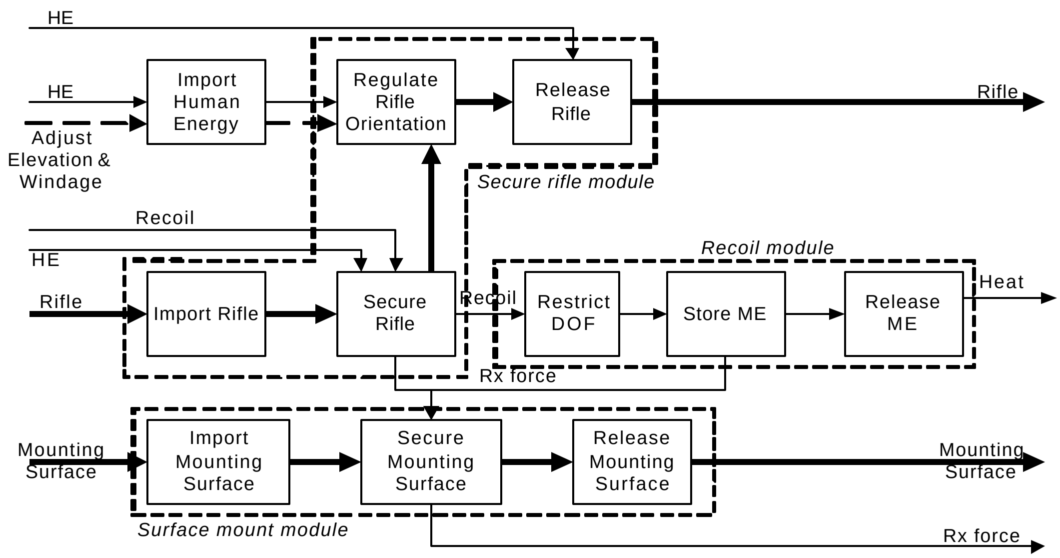


Figure 5.13 Functional model for the machine rest

The functional model shows three modules identified based on the dominant flow heuristic (Stone, 1997).

Table 5.9 Morphological Matrix for the machine rest

Function	Solution 1	Solution 2	Solution 3	Solution 4
Import HE	Handle	Knurled knob		
Regulate Rifle Orientation	Screw adjustment	Shim	Spring loaded detent	
Import Rifle	V-base with pad	Flat plate (buttplate)	Screw into float tube	Upper receiver pins
Secure / Release Rifle	Screw – vise	Cam lock	Pin lock	Screw into barrel float tube
Restrict DOF	Linear bearings	V-blocks	Keyed shaft	4-bar linkage
Store / Release ME	Spring			
Import Mounting Surface	Flat base			
Secure / Release Mounting Surface	C-clamp	Impaling structure (for ground)		

5.4.3 Architecture Design

The following discussion presents the development of multiple alternative architecture solutions to the machine rest design problem. In order to prime the development process, the shaded regions in the morphological matrix represent initially selected physical solutions from which a baseline spatial constraints layout is generated. A total of six spatial layouts are given and one layout is selected for further development at the functional layout level. As with the set of spatial constraints solutions, several functional layouts are explored and one is chosen for further development at the physical solution level. Although only one physical solution diagram is presented, multiple alternatives are explicitly specified within this diagram. One solution from the physical solution is subsequently developed in terms of the remaining three diagrams of the architecture workframe representation. A summary of the set of alternatives follows the presentation of each diagram in the workframe.

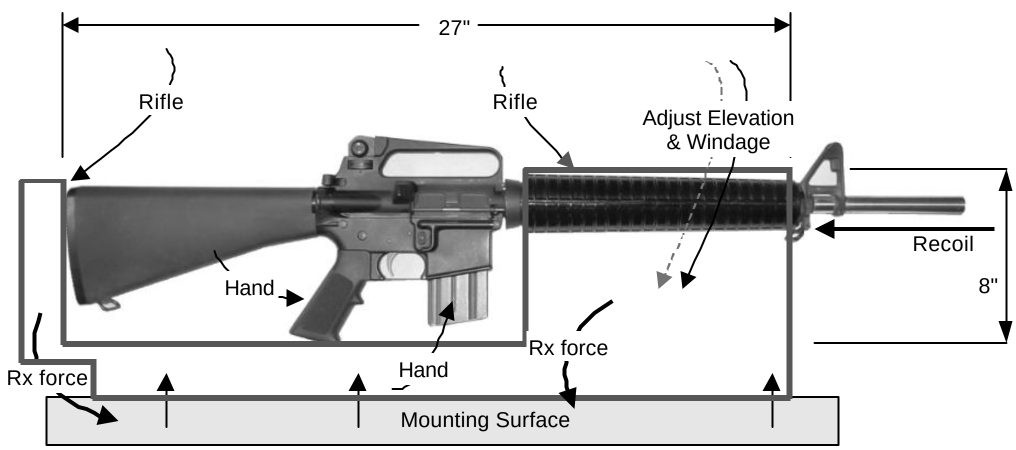


Figure 5.14 Spatial Constraints Layout – Solution 1

The spatial constraints diagram in Figure 5.14 shows that the overall length will be about 2 feet. Device height with the pistol grip exposed is relatively high compared with a bolt action rifle. Additionally, this height causes a considerable moment arm given that the recoil is far away from the supporting structure. Perhaps more importantly, magazine access is somewhat restricted. This indicates that an access region must be available for the user to insert and remove the magazine. If only an AR-15 system is used, one likely layout change is the removal of the rear support structure as shown in Figure 5.15.

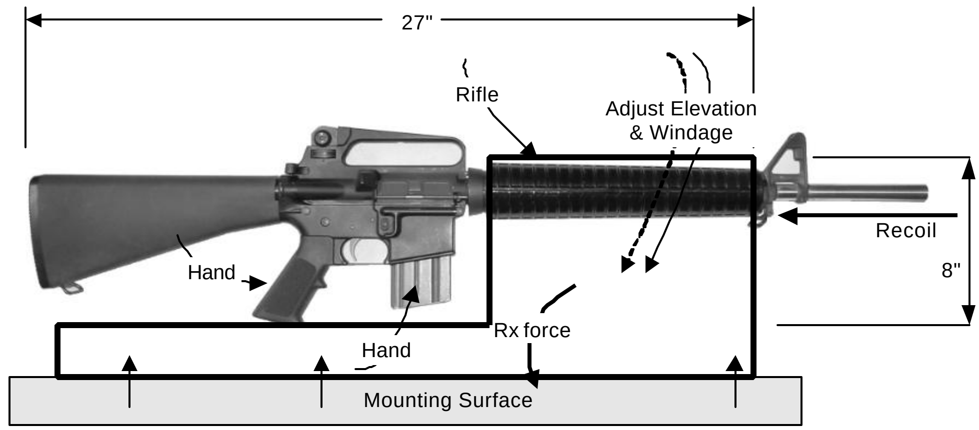


Figure 5.15 Spatial Constraints Layout – Solution 2

In the event that the ground is used as the mounting surface, solution 3 in Figure 5.16 is becomes appropriate.

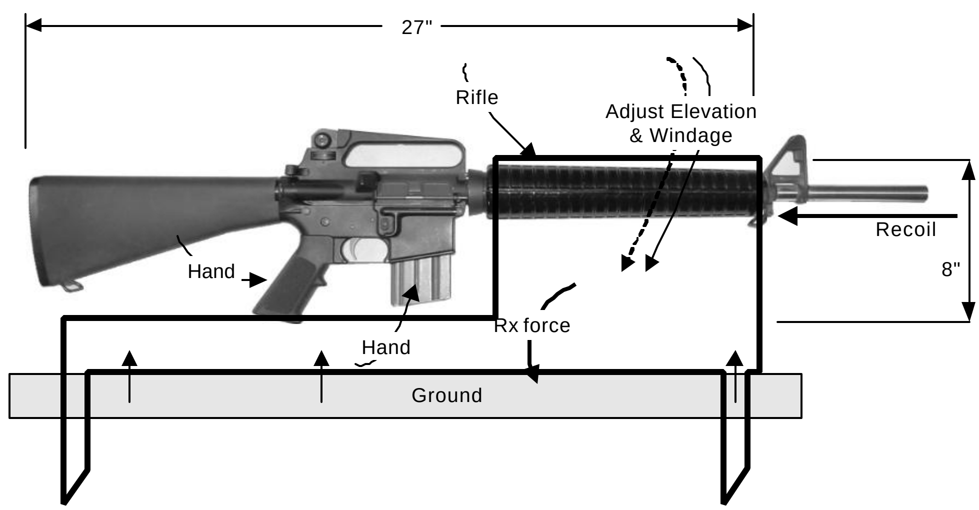


Figure 5.16 Spatial Constraints Layout – Solution 3

Considering the bench mounted case once again, one potential option is to utilize the upper carry handle as a mounting surface since this component already has a hole drilled. Figure 5.17 shows this case.

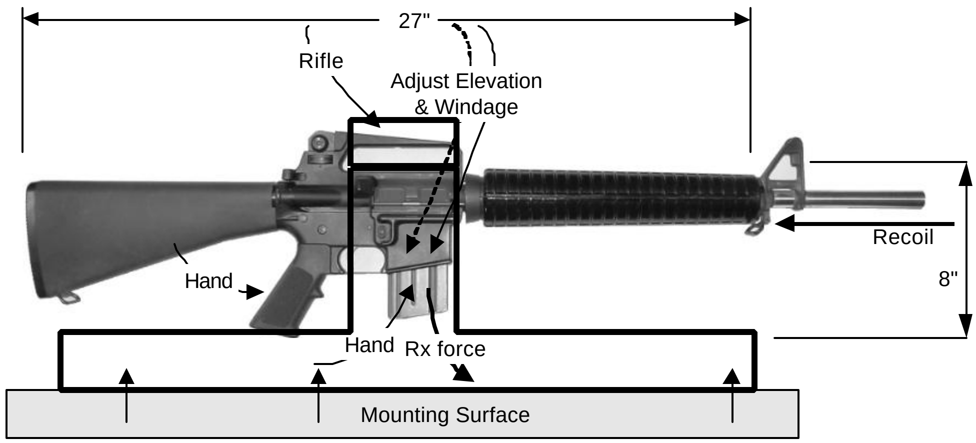


Figure 5.17 Spatial Constraints Layout – Solution 4

If one takes this option to a more extreme level, it suggests that the entire lower receiver can be removed prior to mounting. Since the complete firing mechanism, with the exception of the trigger group, is in the upper receiver, the upper assembly can be mounted using the two main pin holes at the bottom of the receiver. Since the overall size of the device reduces considerably as shown in Figure 5.18, this option certainly has advantages.

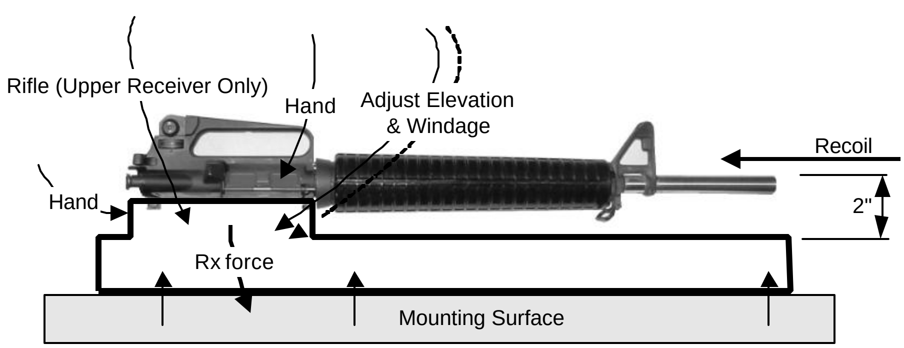


Figure 5.18 Spatial Constraints Layout – Solution 5

Finally, as shown in Figure 5.19, it is possible to support the barrel float tube (assuming a match rifle with a float tube present) directly by drilling and tapping the tube to fit screws on the device.

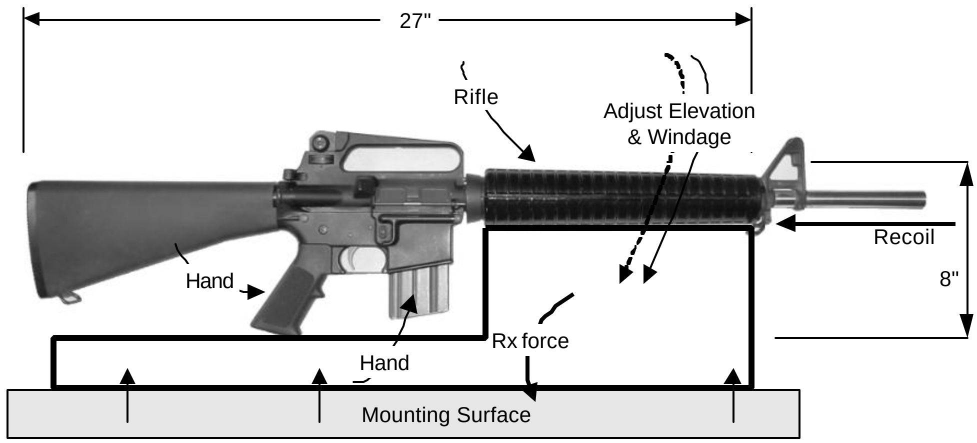


Figure 5.19 Spatial Constraints Layout – Solution 6

Each solution thus far is within the realm of feasibility and each solution has advantages and disadvantages. For purposes of this design exercise, only one solution is pursued based on its overall likelihood of success – solution 2. Figure 5.20 shows the functional layout solution for this spatial constraints layout and indicates each of the three functional modules as identified from the functional model given earlier.

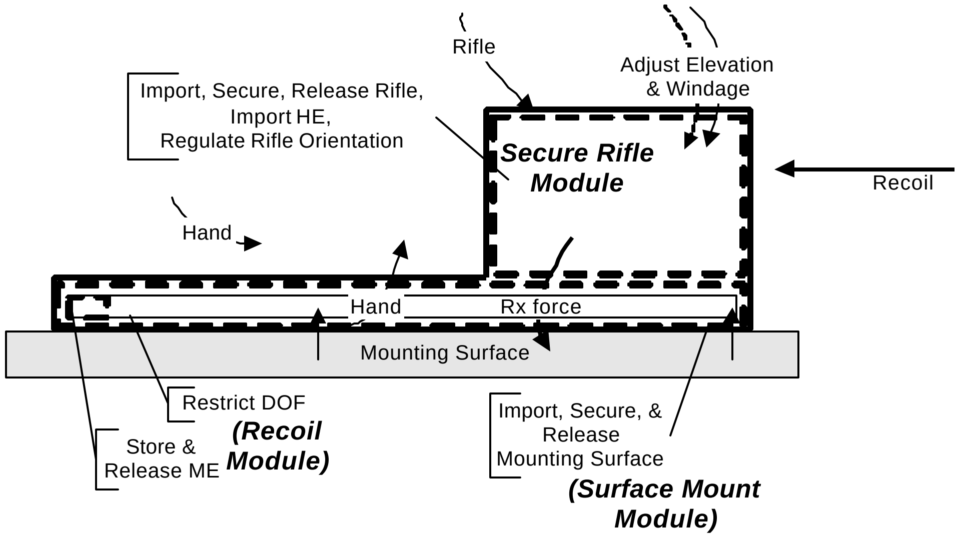


Figure 5.20 Functional layout – Solution 1

Since this solution is based in part on the three modules from the function structure, it is reasonable to explore alternative functional layouts that provide further details and potential advantages to this solution. In Figure 5.21, the functions within the "Secure Rifle Module" are segregated into two regions that distinguish the importation and the regulation functionality.

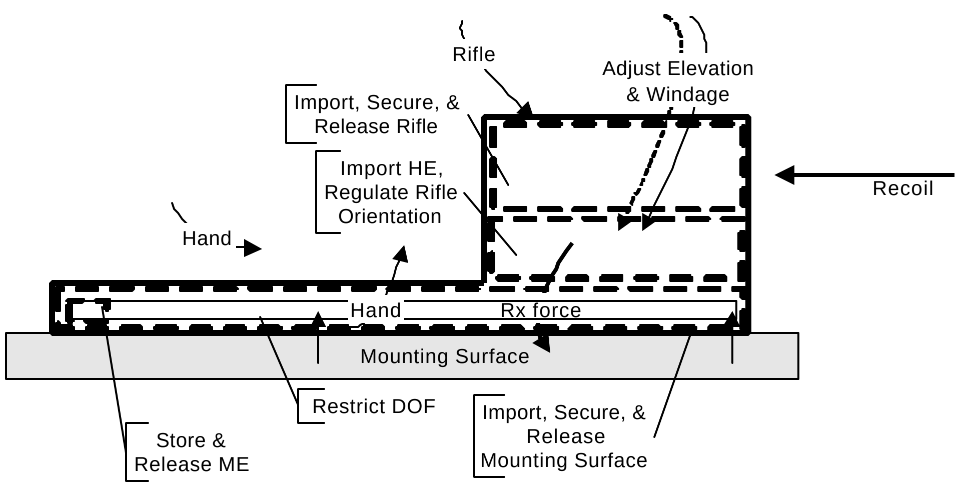


Figure 5.21 Functional layout – Solution 2

Taking this step a bit further, Figure 5.22 shows the regulate rifle orientation (adjust the aim) functionality can also be placed in the “Surface Mount Module” even though the function structure did not identify this as an option.

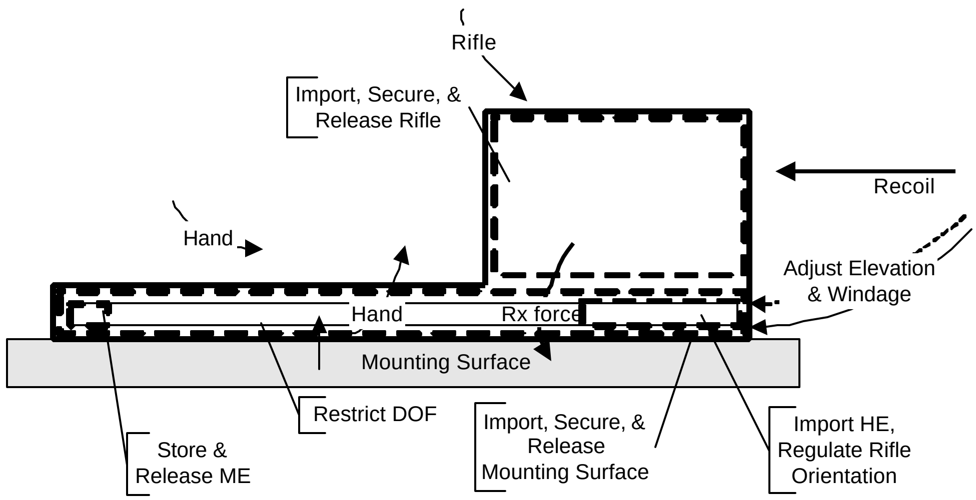


Figure 5.22 Functional layout – Solution 3

Similarly, the “Recoil Module can be moved closer to the rifle rather than positioned near the mounting surface as shown in Figure 5.23.

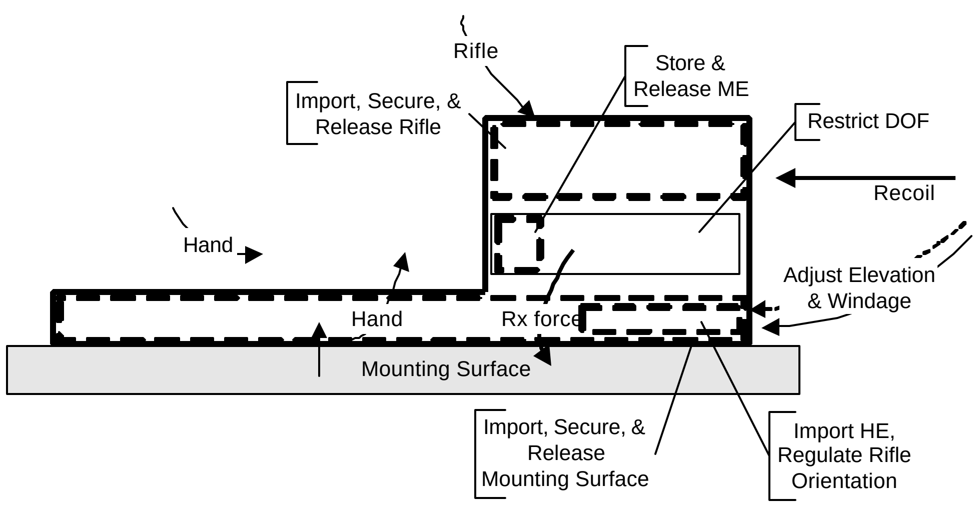


Figure 5.23 Functional layout – Solution 4

If functional layout solution 2 is considered for further development, the following physical solution diagram applies.

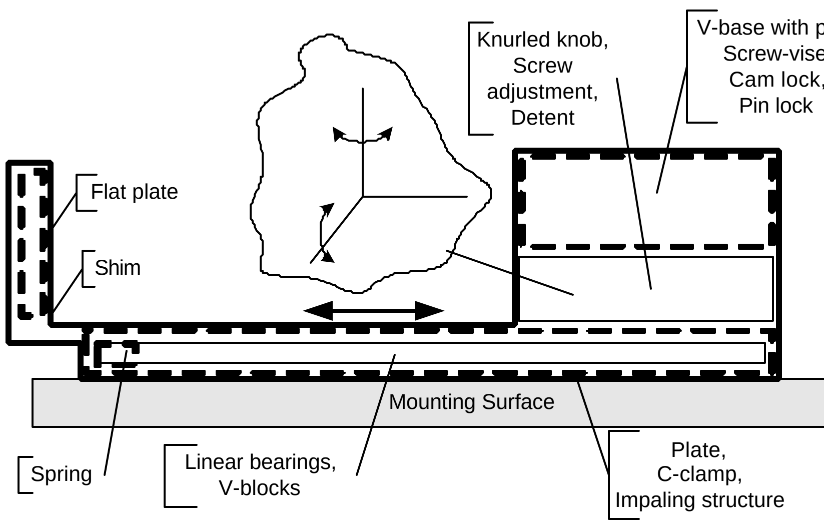


Figure 5.24 Physical Solution Layout

The physical solution diagram in Figure 5.24 shows that multiple alternative solutions can fit within the same spatial constraints diagram. In the event that a solution is grossly different than the size and shape of the product boundary, then a new spatial constraints diagram and possibly a revised functional layout diagram can be developed to accommodate this new alternative in future revisions.

In the partition diagram shown in Figure 5.25, note that the three modules identified from the function structure are present although additional sub-modules are also shown based on estimates of likely manufacturing and assembly options. The partition diagram shows that depending on how many fasteners are present, the device will contain approximately two dozen components.

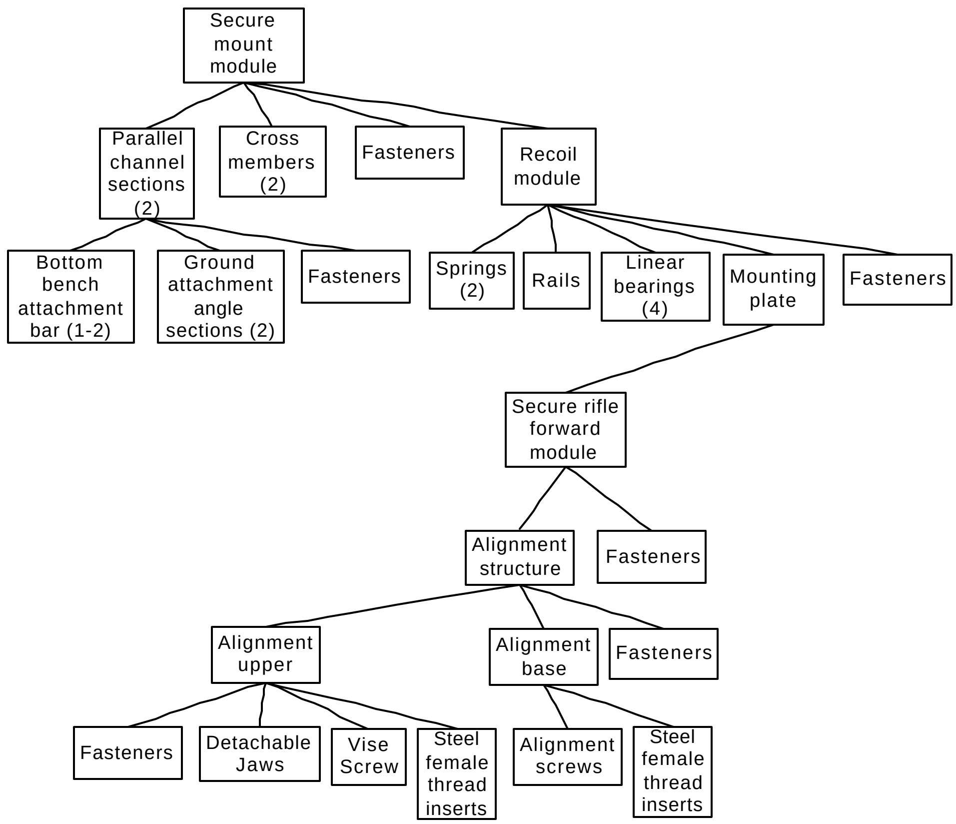


Figure 5.25 Partition Diagram

Given the partition diagram which decomposes the architecture into the majority of all components, the manufacturing diagram in Figure 5.26 below adds mainly material choice information and component sourcing information in terms of either in-house or OEM (Original Equipment Manufacturer).

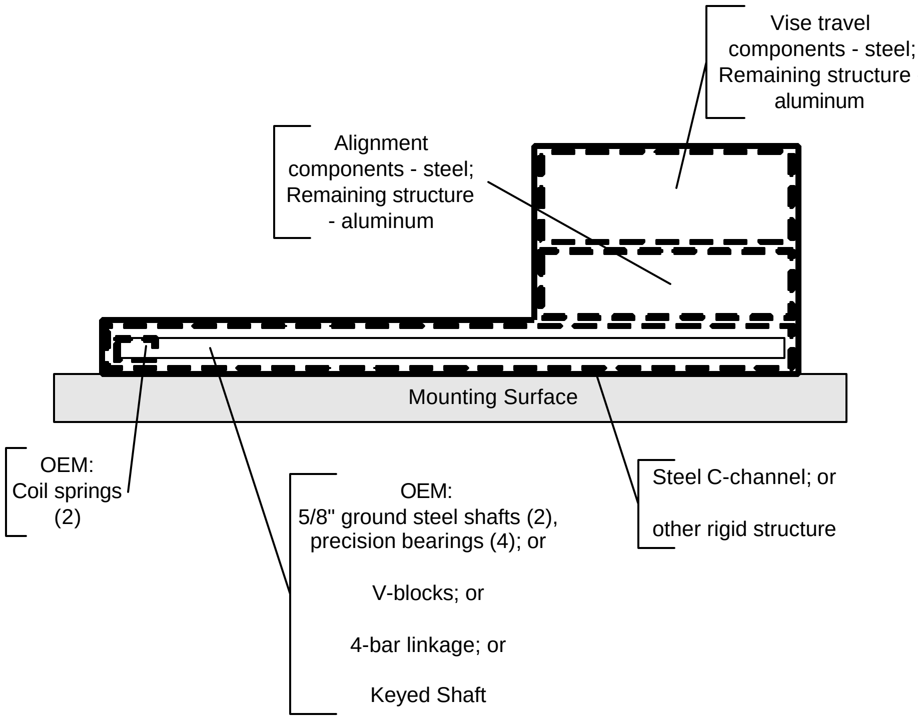


Figure 5.26 Manufacturing Diagram

The product family diagram in Figure 5.27 below shows how the device can be used in different applications, bench and ground use, and for different rifles. In adopting these modular attachments, the bulk of the system is a common platform for the variant configurations.

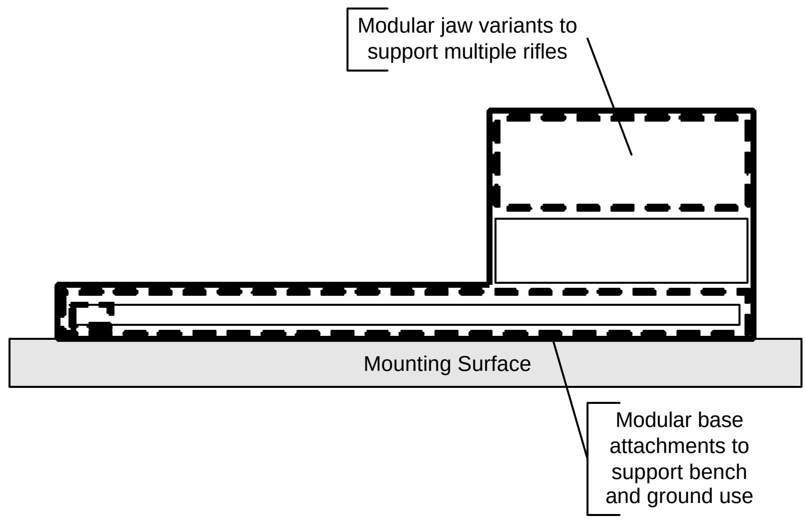


Figure 5.27 Product Family Diagram

Several alternatives are considered above, and Figure 5.28 provides a summary of their development.

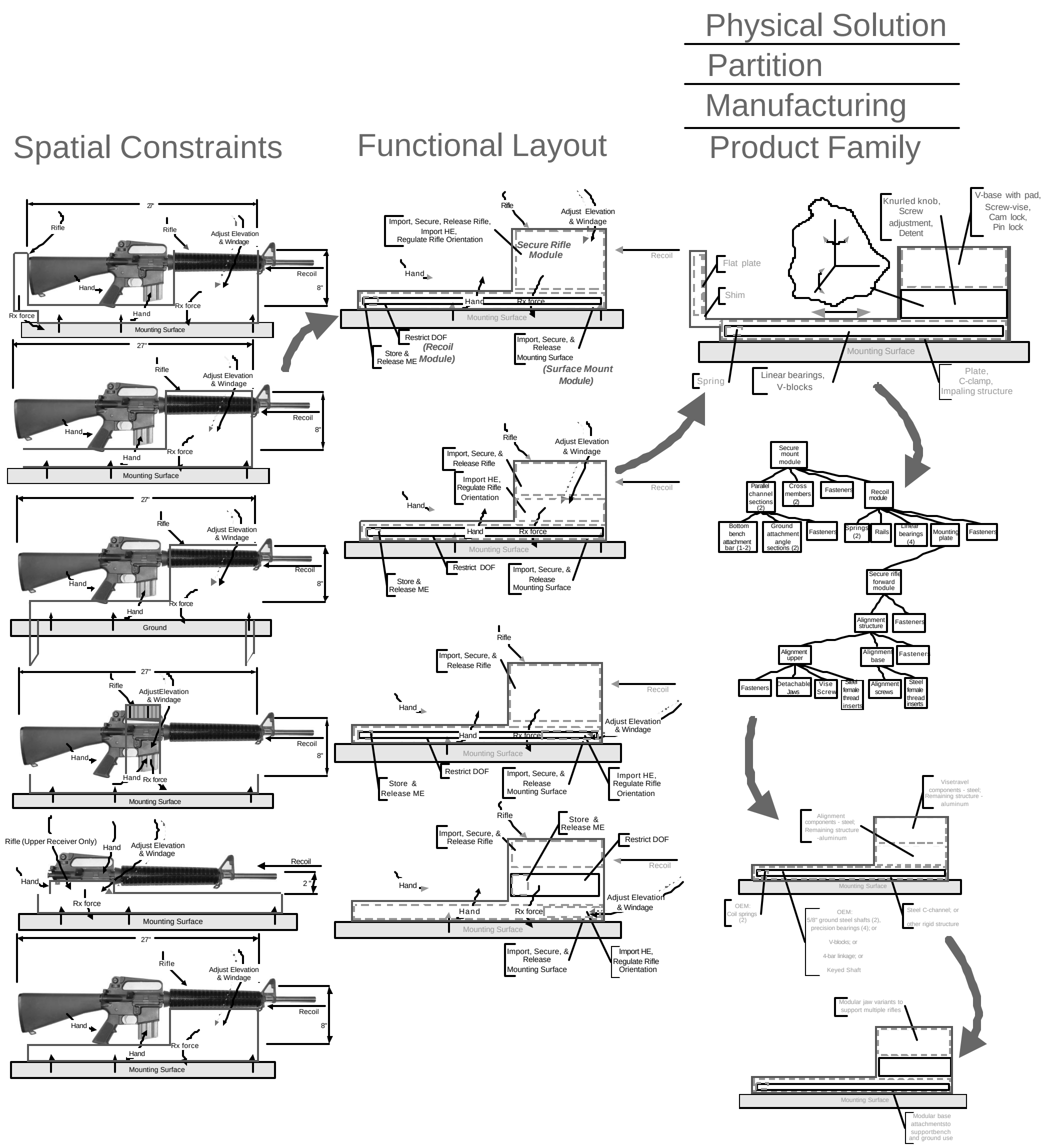


Figure 5.28 Development of alternative architecture layouts

In addition to providing information for the product architecture, the architecture workframe combined with the morphological matrix also indicates potential features with respect to portfolio architecture in the event that this machine rest is ever part of a product family. Table 5.10 provides details about the platform and variant modules in this regard.

Table 5.10 Architecture Platform Alternatives

Shared items among the platform concepts	**Platform #1**	**Platform #2**	**Platform #3**
A frame structure based on two C-channel sections A flat plate as a breadboard for the secure rifle module	Linear bearing based recoil module	V-block based recoil module	Four-bar linkage recoil module

Table 5.11 Architecture Variant Alternatives

Possible modular attachments usable in each variant	**Variant #1**	**Variant #2**	**Variant #3**
Modular base attachment for mounting on ground Modular secure-rifle attachments that are suited for multiple rifles	Full size secure-rifle module that includes an integrated forward and rear unit	Adjustable mount version for convenient regulation of elevation and windage	Rigid forward-only mount

Given the information regarding potential alternatives and the available materials, a concept is developed and prototyped. Results of the final design are given next.

5.4.4 Post-Architecture Design

Based on the second spatial constraints solutions, the second functional layout solution, platform 1 and variant 3, an embodiment of the design is developed, prototyped, and tested. As a result of the prototyping effort, potential improvements in the design are identified although these opportunities do not significantly affect the functionality or architecture and these avenues are left to future work. Figures 5.29 and 5.30 show the machine rest alone and with a rifle.

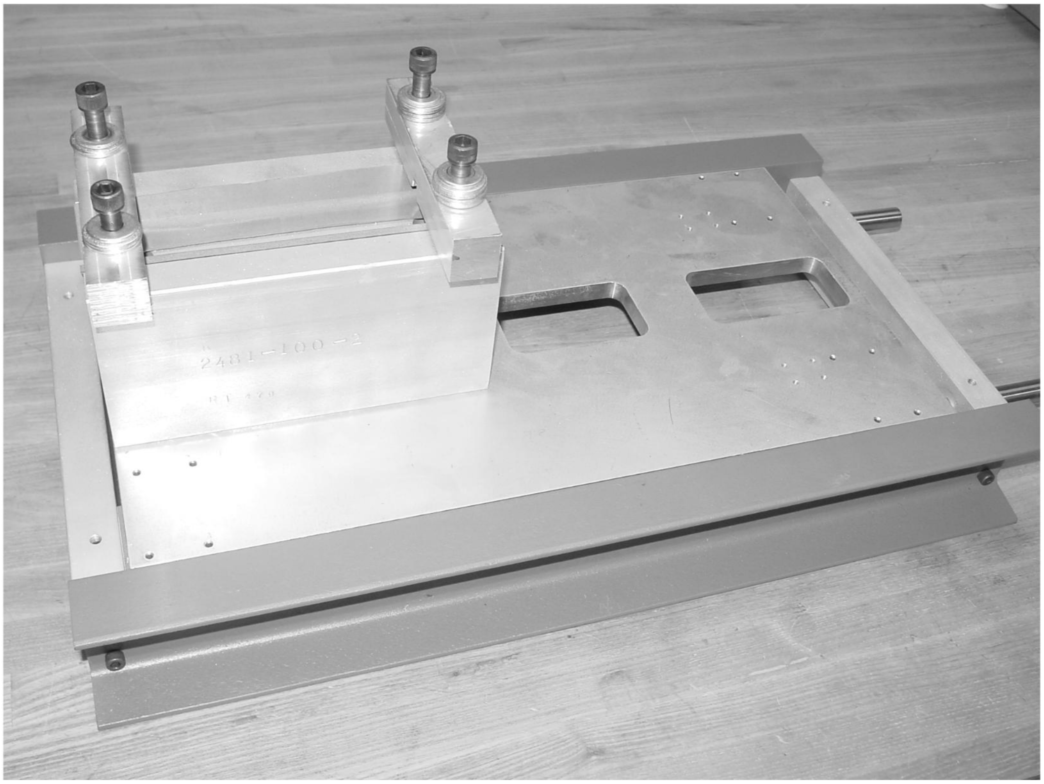


Figure 5.29   Rifle Machine Rest

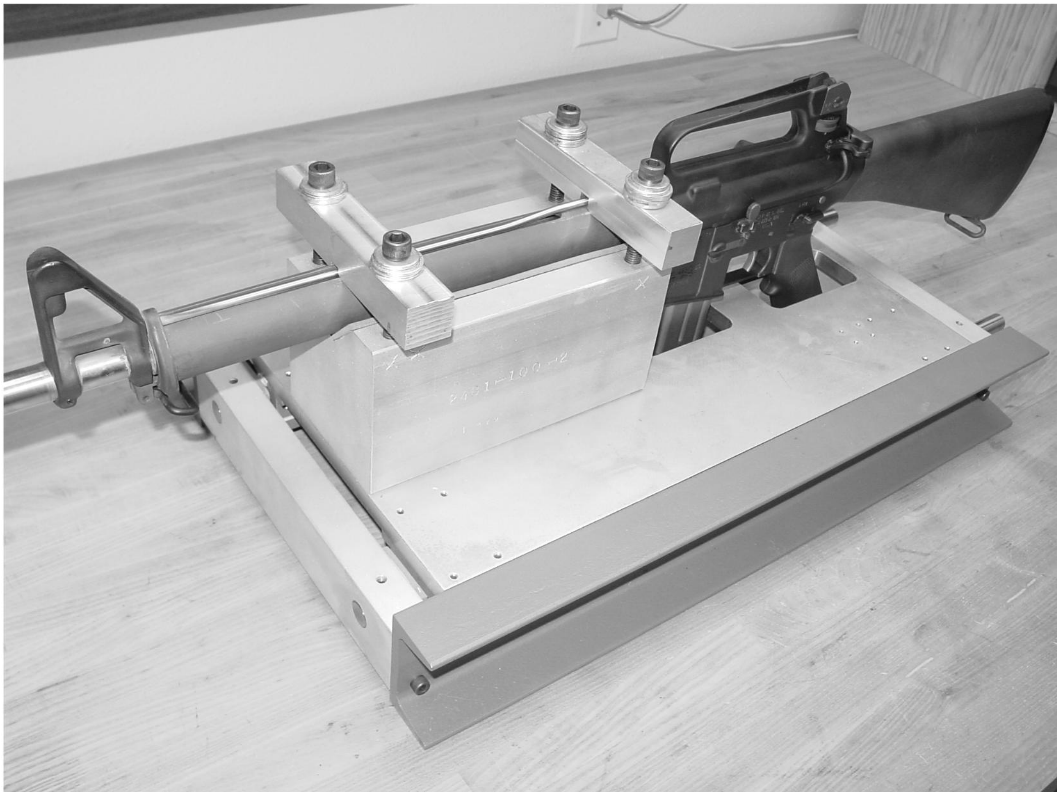


Figure 5.30 Machine rest with rifle mounted

As the figure above illustrates, the fiberglass handguards on the rifle are removed due to their somewhat moveable fit on the upper receiver. Instead of these handguards, the barrel float tube, made from about 1/8" steel tubing, serves as the mounting surface. By using this steel extension of the receiver, a solid mount is achieved. During testing, a five shot group at 100 yards is made using the machine rest and the actual target is shown in Figure 5.31. A US dime and a one inch scale is provided for size comparison. Extreme center-to-center distance for this group is about 0.31 inches or 0.31 Minute of Angle (MOA) which is exceptional for this match grade AR-15. Relative to the levels of precision in the shooting discipline typically associated with this particular system – "NRA High Power Competition," this level of precision is at the high end of repeatability for any such rifle being fired with a human attached to it even under pristine conditions such as firing from a rest or support. Given the ease with which this group was obtained (ie. mount the rifle to the rest and fire without realignment of the sights), it demonstrates that the machine rest offers a great advantage in testing ammunition. The device provides an efficient means for removing human errors normally present in rifle shooting when testing ammunition.

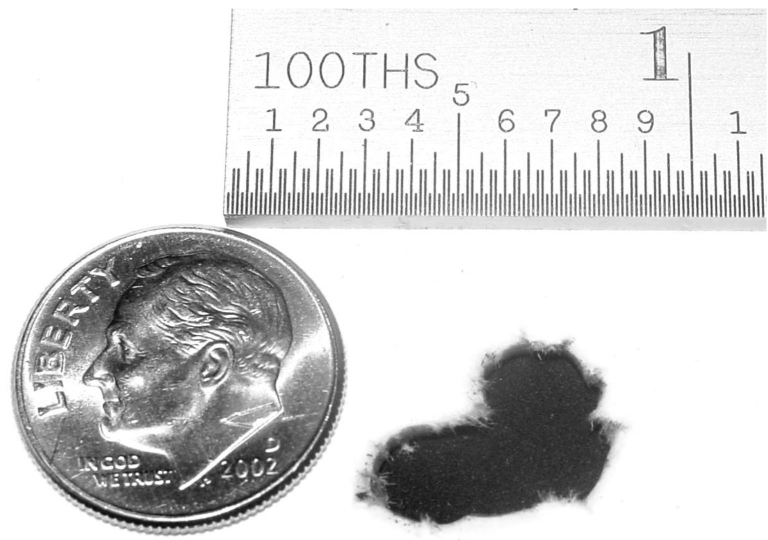


Figure 5.31 Five shot group at 100 yards from an AR-15 (.223 Rem.)

5.5 METHOD EVALUATION

Generally, method validation is described by the observation step from the research model outlined earlier. This is the process used for validation of both the architecture representation and the guidelines. For the case of the method, an additional validation scheme is imposed on this observation step in order to more thoroughly show support for the method. This extra step is based on the Validation Square approach which is designed for the specific purpose of validating design methods (Pederson et al., 2000). The following discussion begins with observations of the method with respect to the requirements and concludes with a test of the method using the validation square.

5.5.1 Observations

Table 5.12 shows the results of comparing the method against the same requirements developed earlier. When making the observations, several reasonable assumptions are imposed on the operating conditions. The first assumption is that the designers using the method have a working knowledge of the method through some preparation. This is to suggest that the method requires some practice, initial use, and familiarization before one can expect a reasonable level of effectiveness. The method is not a 'plug and play' tool like a new printer that works at maximum performance immediately after installation. This is not an unusual burden and generally speaking, the same assumption is true for other advanced methods (Urlich and Eppinger, 2000; Pahl and Beitz, 1996; Otto and Wood, 2001). This dissertation does not claim that this method requires fewer setup man-hours, but instead that this method is relatively effective once the designer has a working knowledge of how to use it. An approximate comparison for the preparation time required to be effective with this method is the functional modeling language. Based on past experience with this language and use of the architecture method, they both require minimally a few hours on the part of the designer to become familiar with the design technique.

The second assumption on method operation is with respect to the kinds of problems that are appropriate for method use. The method is intended to be used for designs that are mostly mechanical and generally medium in scale. As mentioned earlier, this excludes the very large and very small domains such as aircraft or MEMS designs.

The third assumption addresses the user. This target customer range is from undergraduate engineering freshman to designers in industry with several years of experience. Generally, the method is geared toward individuals and smaller teams (1-5 people). Although in principle larger teams may elect to use the method, the overhead of keeping track of design solutions may become a significant difficulty with large numbers of users directing action. Given that the above assumptions are in effect, the following assessment reflects method performance.

Table 5.12 Assessment Summary

Practical, Easy to Use	The method does not require an unusually lengthy preparation time in terms of either a learning curve or the extent of resources required to implement the technique.
Complete and Comprehensive	The method contains a means for representing architecture and for guiding the designer toward good solutions.
Interface well with the overall design process	The method has clear inputs and outputs to the remainder of the design process.
Support generation of alternative architecture concepts	The method is effective in directing the designer to alternatives. This result is due largely to the representation which decomposes the architecture problem into steps that individually may yield several alternatives.
Systematic	The method provides explicit and incremental steps to direct the designer from function to form.
Be robust to typical design project noise	The method should function despite typical project noise since the assumptions regarding minimum required inputs to the method are not unusually high.

5.5.2 Validation Square

The above observations give a general overview of the method's capabilities. The following discussion treats the method in greater detail and incrementally addresses method's validity by beginning with statements about method constituents and leading toward more general claims of method performance. The 'validation square' is a technique designed to assess a method in a series of six steps as shown in the lower portion of Figure 5.32 (Pedersen et al., 2000).

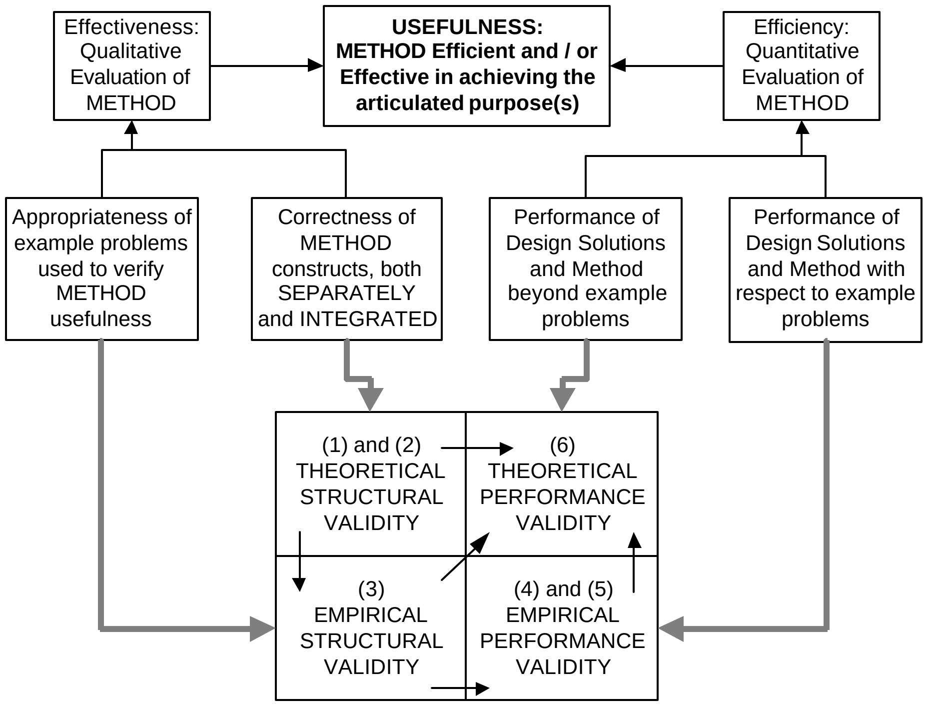


Figure 5.32 Validation Square (Pedersen et al., 2000)

The first step is to show validity of the individual constructs that makeup the method. In this case, the major components of the method are its inputs, outputs, the representation workframe, the innovation process module, and the guideline module. The inputs and outputs are addressed earlier in this chapter and are satisfactory with respect to typical descriptions of the design process found in the literature (Pahl and Beitz, 1996; IEEE Computer Society, 1998). Based on the thorough development of the representation in Chapter 3, the representation is considered to be valid and appropriate for the method. The same is true for the guidelines which are individually addressed and validated in Chapter 4. While many candidate creative methods exist, the innovation process module is based on those which are well established and therefore this portion of the method is considered acceptable as well.

The second step in validating the theoretical structure of the method is an examination of method consistency. The task is to ensure that the method process is fundamentally feasible in terms of having sufficient information to execute method steps. Additionally, this check determines if information generated is either excessive, unnecessary, or even invalid. Each of the three method steps was described by the step task, the inputs required, and the outputs achieved. These are given in Tables 5.5, 5.6, and 5.7. Based on these descriptions it is clear that sufficient information exists to carry out each step. These descriptions also show that each step produces a reasonable output that is certainly not excessive or invalid. The guidelines are an exception to this last statement to some extent. Not all guidelines are appropriate for all situations and naturally the valid use of the guidelines is dependent on the designer's judgment in applying the guidelines to an appropriate problem in proper context.

The third step is in addressing the empirical structural validity and this means accepting the example problems that are used to verify method performance. The method is designed for a range of problems that are mostly mechanical and at the small and medium scale. Given this type of application, both the rifle machine rest example and a nail gun example, which will be discussed shortly, are similar and fit in the same category.

The fourth step in the validation square is accepting the usefulness of the method for some example problems. That is, does the method actually help the situation given the example problems that are considered acceptable from the previous step. This is where the validation square and the research model in Figure 2.1 overlap somewhat. In both cases, the method is measured against some relevant metrics. Given this commonality, the assessment summary from Table 5.12 directly supports the method in this step. In addition to these observations, results from the machine rest problem and the upcoming nail gun example problem, discussed next, also demonstrate method utility.

For the nail gun problem, an experiment is performed to test the method relative to another method. A portion of the method developed in this dissertation is compared with one other design method that is specifically intended to aid in designing product architecture: the method proposed by Ulrich and Eppinger (2000). As highlighted in the introduction, this method is a four-step process as described below in Table 5.13. This particular technique is chosen for comparison since it is targeted toward architecture design and considered to be at least as good as the nominal level of capability for conventional engineering practices used by engineers today.

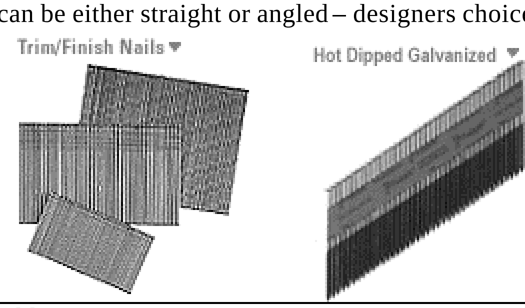
Table 5.13 Architecture Design Method from Ulrich and Eppinger (2000)

1. Create schematic containing elements of both function and form.
2. Cluster elements of the schematic
3. Create a rough geometric layout
4. Identify fundamental and incidental interactions

An experiment is run to test the differences between the architecture design method and the Ulrich and Eppinger (2000) technique. The experimental setup is developed to test method utility with respect to three metrics: 1) the quantity of concept alternatives produced, 2) the quality of the solutions, and 3) method efficiency or the quality per the amount of time allocated to each concept. The experiment is run as an in-class assignment for senior mechanical engineering design students at the University of Texas at Austin. The students are divided into two person teams and two groups are formed using these teams. A control group is given the conventional technique while the experimental group is given a portion of the architecture method from this dissertation. Due to limited time resources on the part of the students, the full entire method from beginning to end is not tested. Only concept development using the first three diagrams in the architecture workframe representation is performed. The following describes the task required by the test and experimental groups.

One major concern is uniformity of conditions between the two groups. Step one of the method includes the generation of a morphological matrix. This step is excluded from the experiment so that the experiment is not testing the ability to generate physical solutions to a set of functions. The main test is with respect to the layout of physical solutions. Both groups are given the same preliminary information describing the problem. This includes customer needs, an activity diagram, a black box, a functional model, and a morphological matrix containing physical solutions to each function in the functional model. Table 5.14 and Figures 5.33 through 5.35 illustrate each of these problem elements.

Table 5.14 Customer Needs for a nail gun

Low Cost	(< $500)
Light weight	(< 6 lbs)
Utilize 18 Gauge finishing nails	(capacity of 100) (1.5” long – only 18 Gauge nails supplied by a single manufacturer such as Paslode) Nails come in glued together in a strip of 50 – the strip can be either straight or angled – designers choice) 
Small size	Able to use in corners No greater than 12” in any dimension
Portable	Can carry on a 1.5” wide belt
Safe	Unable to fire by dropping on ground Unlikely to be fired by children
Last long before energy resupply	Efficient use of energy
One handed nailing operation	Should be able to support firing using one hand at any orientation (ceiling, floor, wall, tight corner, etc.)

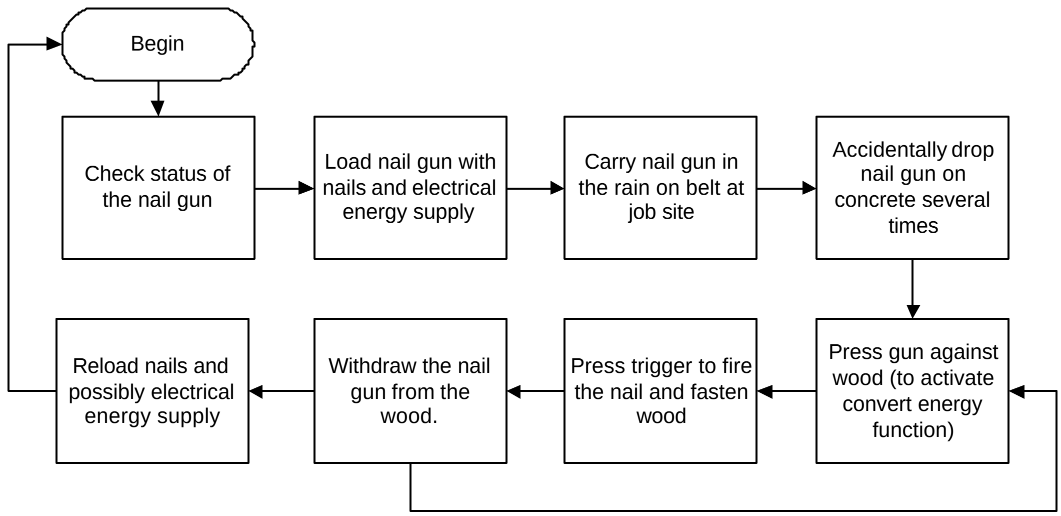


Figure 5.33 Activity Diagram for a nail gun

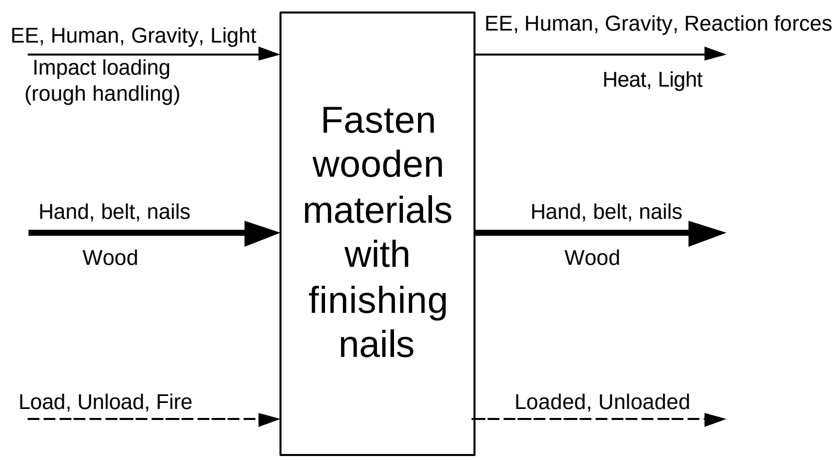


Figure 5.34 Black Box for a nail gun

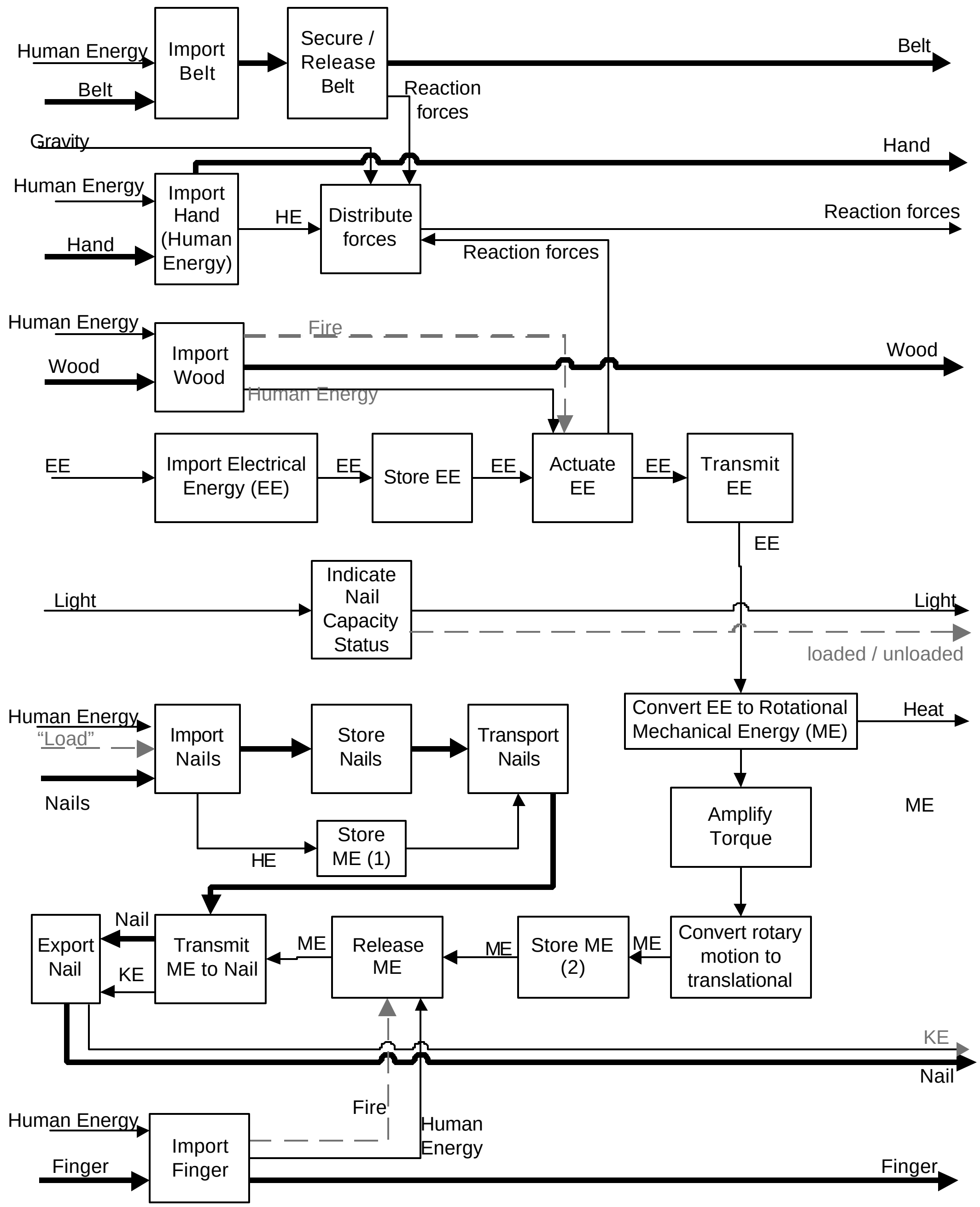


Figure 5.35 Functional Model for a nail gun

Given the above problem information, both groups are organized into two person teams and then separated and briefly instructed on the use of their assigned design method. Following this brief introduction to the method, each group works for approximately one hour to generate alternative concept solutions for the nail gun problem. From the student's perspective, the assignment objective is to be graded on the quantity and quality of the results in addition to following the design procedure set forth by the design method. For the control group, all four steps of the method defined in Table 5.13 are specified. For the experimental group, the assignment covers development of concepts based on three diagrams in the representation workframe: the spatial constraints diagram, function layout diagram, and the physical solutions diagram. These three are chosen since they accomplish roughly the same function as the process prescribed by the conventional technique. Additionally, the limited time available for the student exercise requires such a reduction in task.

The results are encouraging as they show distinct differences between the control and experimental groups. As expected, the experimental group generally demonstrates a greater degree of problem decomposition afforded by the architecture representation. Figures 5.36 through 5.38 show the development of one concept layout from one team in the experimental group in terms of spatial constraints, functional layout, and physical solutions.

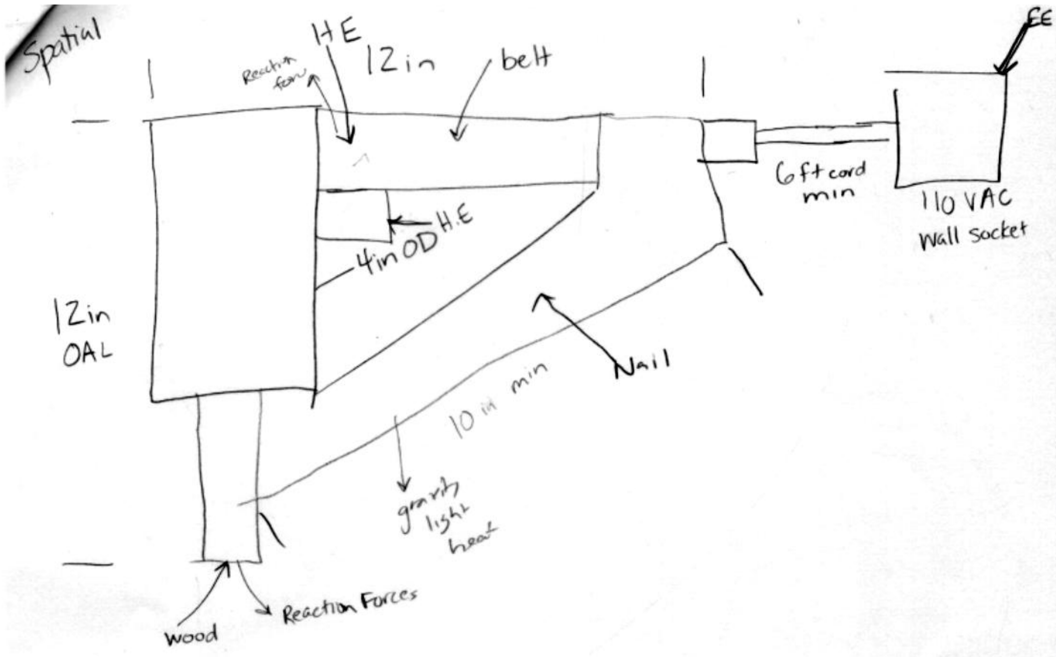


Figure 5.36 Spatial Constraints Diagram – Experimental Group

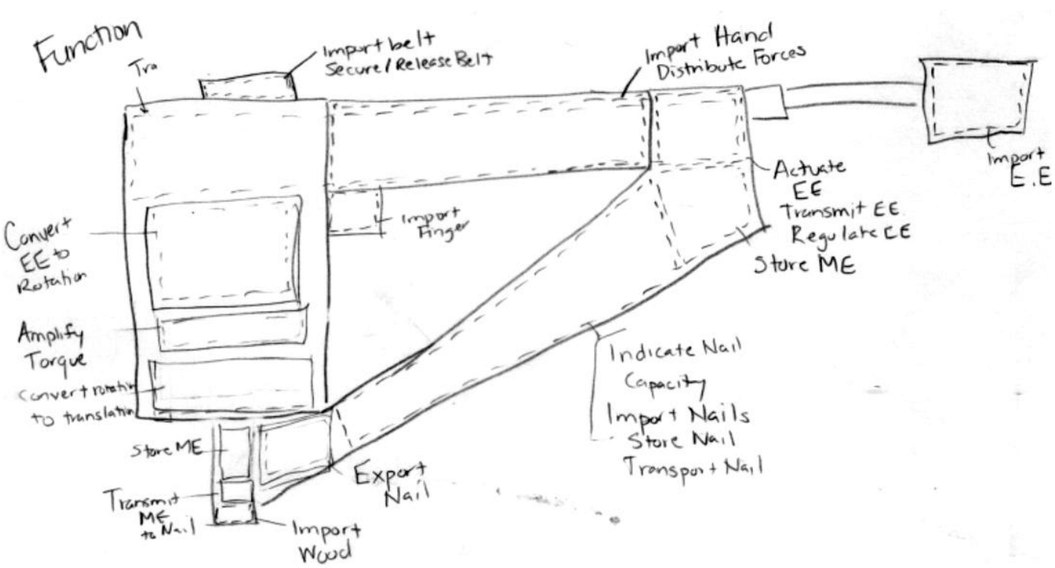


Figure 5.37 Functional Layout Diagram – Experimental Group

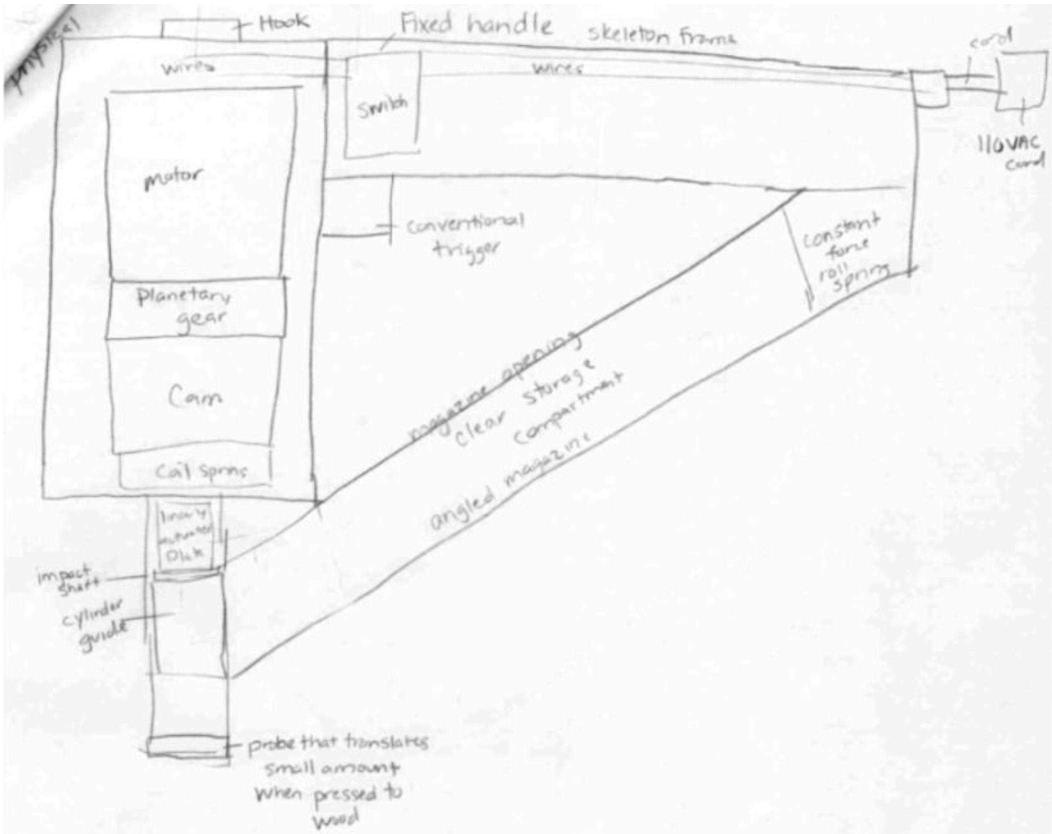


Figure 5.38 Physical Solution Diagram – Experimental Group

A second team in the experimental group demonstrates a considerable degree of concept exploration through the generation of multiple spatial constraints diagrams as shown in Figure 5.39. Additionally, this same team takes advantage of the physical solution diagram by explicitly showing alternative candidate solutions within the physical solution diagram given in Figure 5.40.

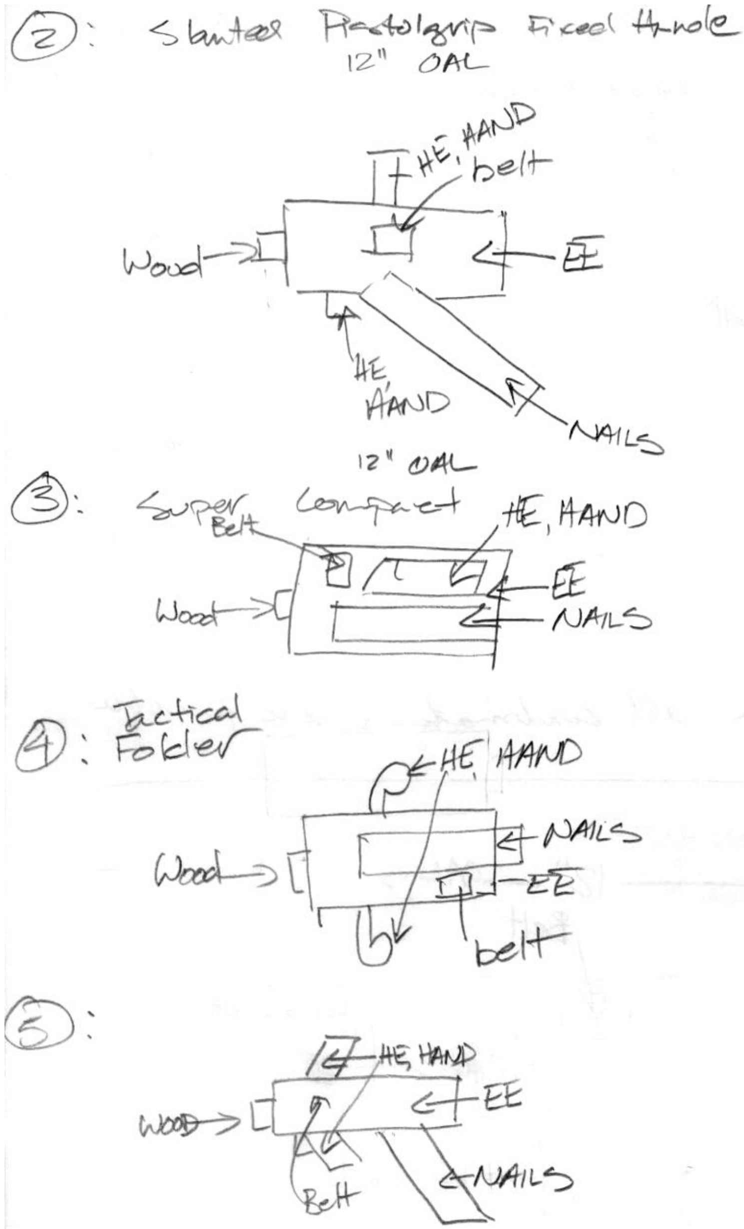


Figure 5.39 Alternative Spatial Constraint Layouts – Experimental Group

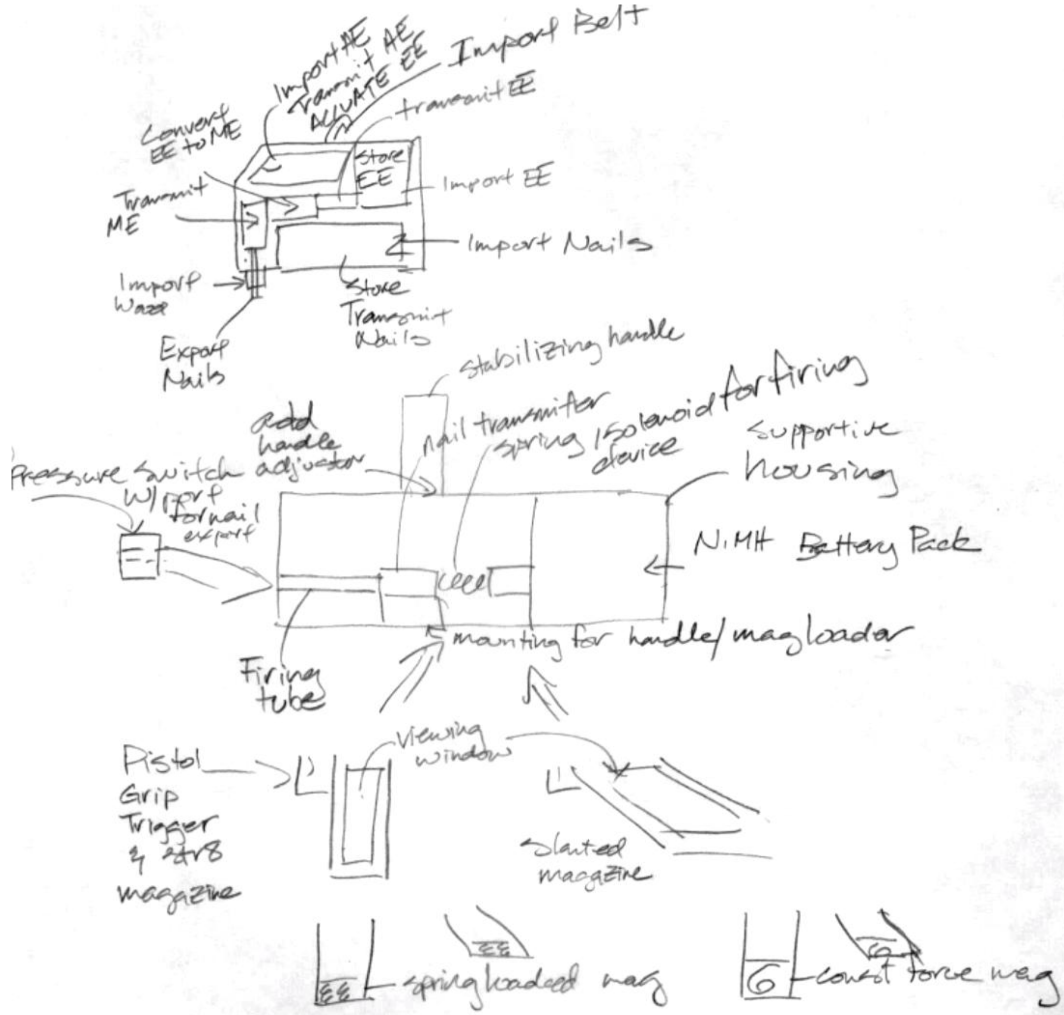


Figure 5.40 Alternative Physical Solutions – Experimental Group

In contrast to the experimental group, the control group exhibits a greater degree of merging of function and form within the same concept. The schematic layout shown in Figure 5.41 illustrates this point. Despite the fewer number of alternatives generated by the control group, the quality of solutions in terms of the geometric layouts were relatively good as illustrated in Figures 5.42 and 5.43.

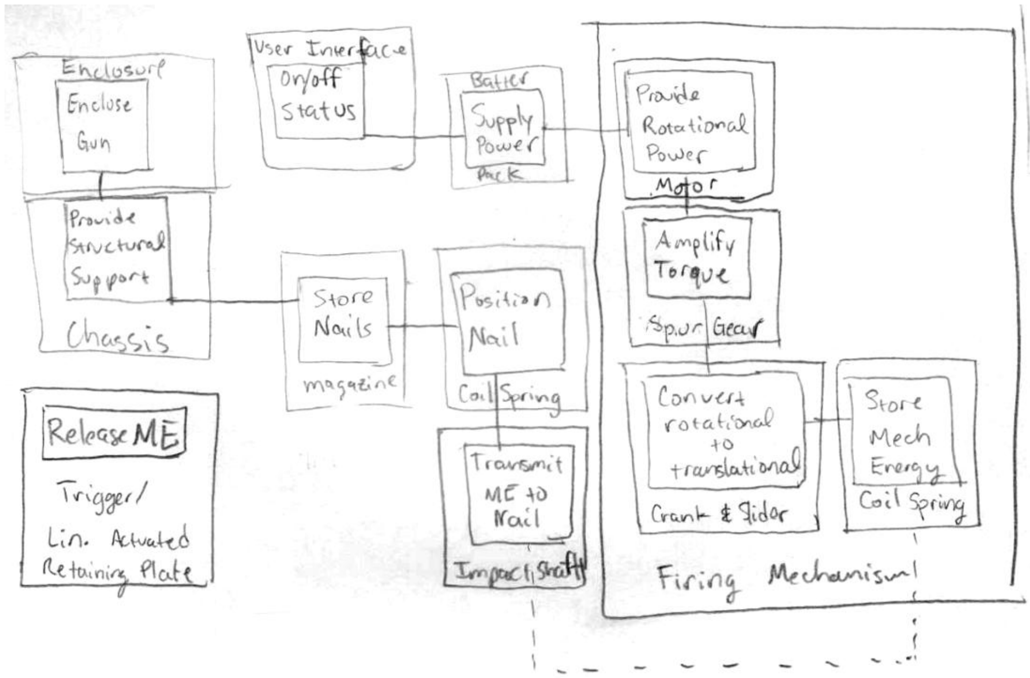


Figure 5.41 Schematic Layout – Control Group

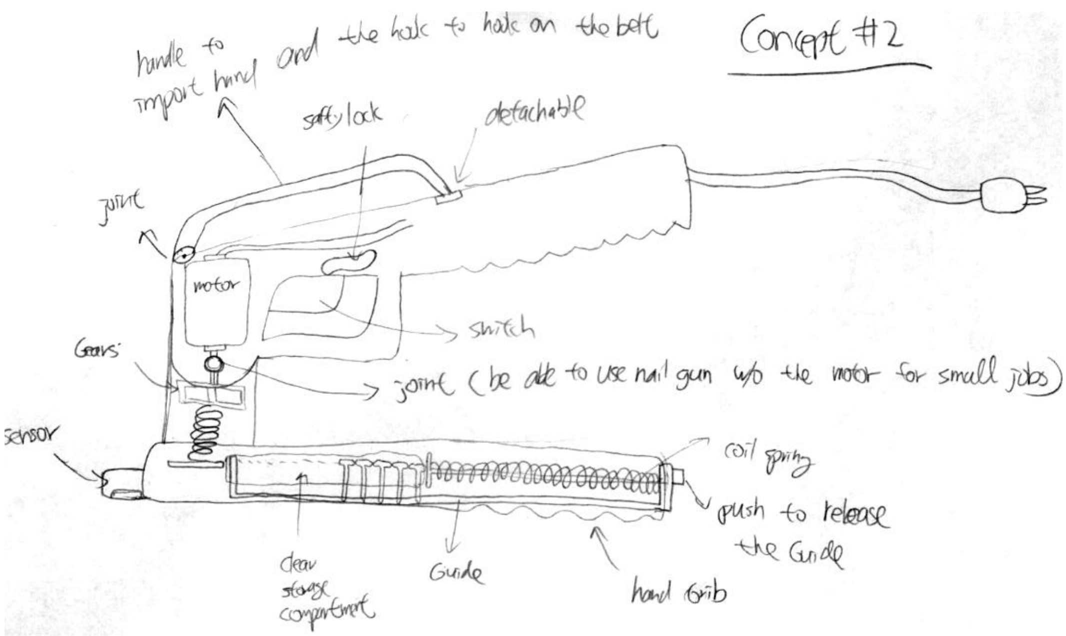


Figure 5.42 Geometric Layout – Control Group

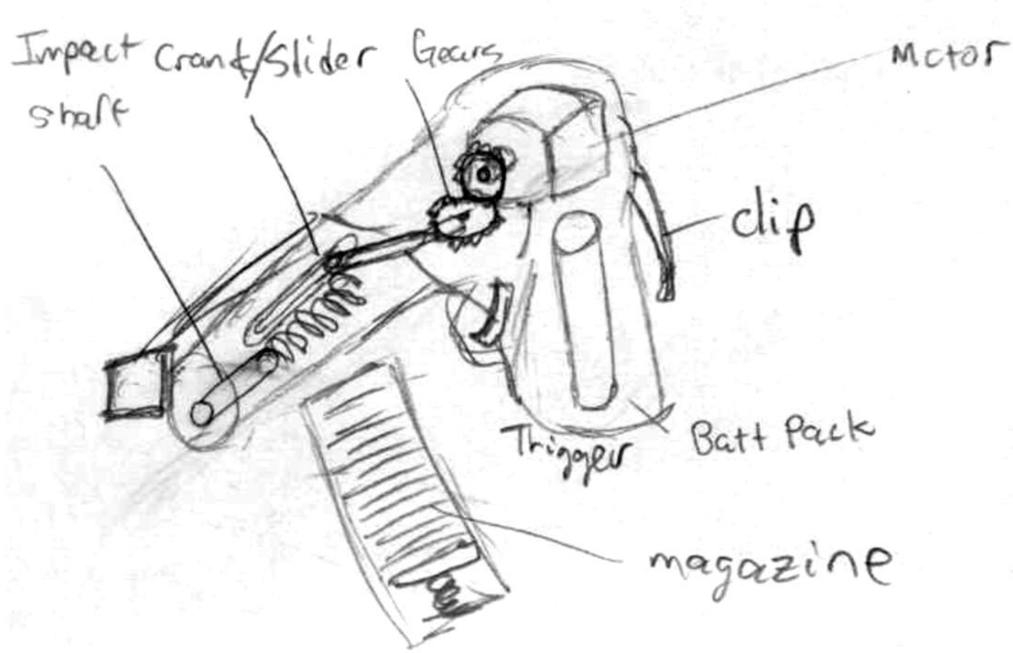


Figure 5.43 Geometric layout – Control Group

The results are evaluated according to the three metrics introduced above: quantity, quality, and method efficiency. Quantity is defined as simply the number of alternative concepts generated. Quality is defined as the worthiness of the concept with respect to customer needs. Method efficiency attempts to capture the "bang to buck" ratio of implementing the methods. This third metric is prompted by the inadequacy of the quantity and quality measures alone. Specifically, if one team generates twice as many solutions as another, then the amount of effort expended per solution varies between the two groups. The method efficiency measure is defined as the quality of solutions per the percent time allocated to each solution. If one makes a reasonable assumption that the time allocated to each concept is the total time divided by the number of solutions, then the percentage of time allocated to each concept is (1/quantity). Therefore the method efficiency metric is defined as (quantity)*(quality). Results from the nail gun experiment are given in Table 5.15.

Table 5.15 Results from the nail gun design assignment

	TEAM	Quantity	Quality (Q) 1	2	3	Ave Q	Method Efficiency (M.E.) 1	2	3	M.E. Ave
CONTROL	1	6	5.0	3.8	5.7	4.8	30.0	22.8	34.2	29.0
	2	1	5.0	5.0	7.0	5.7	5.0	5.0	7.0	5.7
	3	3	8.3	6.0	8.0	7.4	25.0	18.0	24.0	22.3
	4	2	6.7	6.8	6.0	6.5	13.3	13.5	12.0	12.9
	5	2	5.0	7.5	9.0	7.2	10.0	15.0	18.0	14.3
	6	2	8.3	8.3	8.0	8.2	16.7	16.5	16.0	16.4
	7	2	8.3	8.0	10.0	8.8	16.7	16.0	20.0	17.6
MEAN		2.7	6.4	6.2	7.3	6.6	16.7	15.1	18.5	16.8
EXPERIMENTAL		**Quantity**	1	2	3	**Ave Q**	1	2	3	**M.E. Ave**
	8	5	6.7	4.4	4.4	5.2	33.3	22.0	22.0	25.8
	9	6	3.3	2.2	4.6	3.4	20.0	13.2	27.6	20.3
	10	8	6.7	3.5	4.9	5.0	53.3	28.0	39.2	40.2
	11	5	5.0	2.2	5.0	4.1	25.0	11.0	25.0	20.3
	12	5	3.3	2.6	5.0	3.6	16.7	13.0	25.0	18.2
	13	5	8.3	7.3	5.2	6.9	41.7	36.5	26.0	34.7
	14	2	10.0	6.0	9.5	8.5	20.0	12.0	19.0	17.0
MEAN		4.3	6.7	4.5	5.5	5.2	30.0	19.4	26.3	25.2
t-test (%)		**97.1**	**13.3**	**94.8**	**100.0**	**86.0**	**93.8**	**64.1**	**87.1**	**89.9**

Since quality is a relatively subjective factor, three people including the author judged the results according to quality. Note the results for the quality of solutions are shown under the "1, 2, 3" headings which indicate the three people evaluating this metric. Note that the graders were not blind to the control and experimental groups. An average quality is also given as the mean average of these three ratings.

The data is compared in terms of the three metrics directly and by performing an unpaired t-test on the means between the control and experimental groups. This test shows the probability that the means are distinct and quite high probabilities are found. Clearly, there is high confidence that the proposed method generates a larger number of alternative concepts than the control method. The quality of solutions varies considerably depending on the evaluator most likely because judging designs is quite subjective. However, the control group does exhibit a higher overall quality probably because teams in those groups generally spent more time per concept solution. In terms of method efficiency, the results indicate that the new method shows improvement over the representative conventional technique. Figures 5.44 through 5.46 show the same results of the three metrics graphically as a function of each team in the study. Figure 5.44 visually shows that the experimental group overall produced a greater number of solutions. As one can see from Figure 5.45, the quality of solutions produced by the control group is generally higher. However, Figure 5.46 shows that the experimental group demonstrated a higher level of efficiency in developing design solutions.

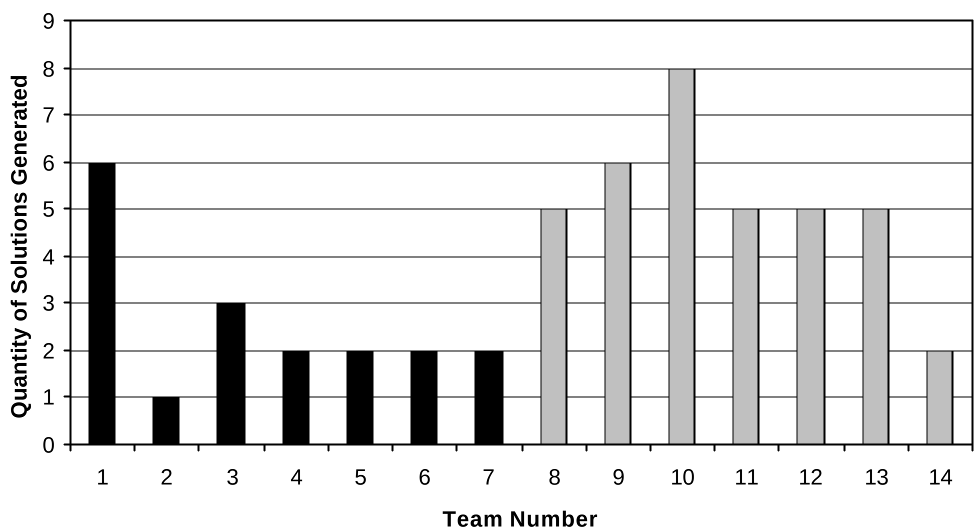


Figure 5.44 Comparison of solution quantity between test groups

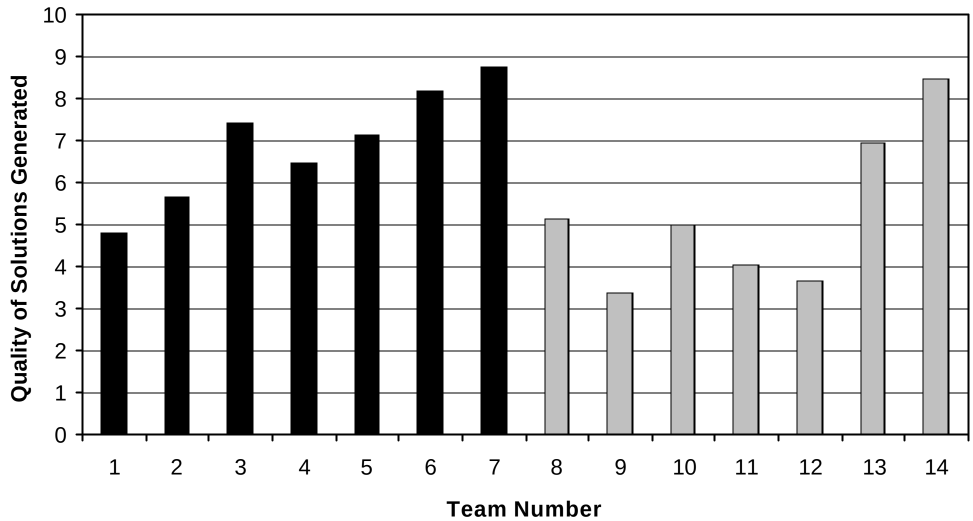


Figure 5.45 Comparison of mean solution quality

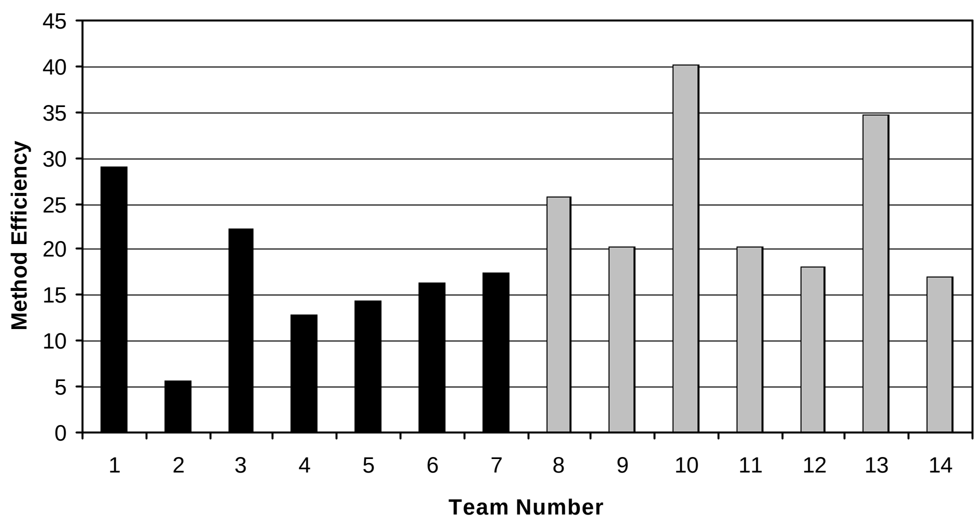


Figure 5.46 Comparison of mean method efficiency

Based on the above results, the method is clearly useful relative to conventional techniques which leads to the fifth step in the validation square process. This is to attribute the usefulness described above as a direct result of implementing the method. Given the controlled conditions of the experiments, the variation between control and experimental groups is attributed to the difference in the two design methods. One potential source of error that may tend to negate this assertion is the variability with respect to rigorously following method procedure among the design teams. Both groups exhibited some deviation from procedure probably due to a lack of extensive practice with either method. Additionally, the composition of the teams between the two groups may differ somewhat with respect to designer skill level and experience. Given that seven teams were involved in each group, the results are encouraging.

The sixth and final step in the validation square process is the acceptance of method utility to cases beyond the example problems. Given the success of the two experiments in addition to the utility toward the original design presented earlier, the leap of faith in claiming a more general degree of usefulness is reasonable. The portion of the method tested in this nail gun experiment is shown to function well in the case of senior design students and this strongly suggests that the method is working as intended. As for the remaining constituent elements of the method not tested in this particular example study, they are validated according to the relevant steps in the validation square as discussed in the early portion of section 5.5.1.

5.6 SUMMARY

This chapter presents a new architecture design method based on the architecture representation and guidelines developed in the previous two chapters. The method is a three-step process that systematically guides the designer through an architecture development process. Two potential initial conditions are considered: original design and redesign. In the original design case, the method begins with the development of physical solutions to satisfy some predetermined set of functionality. In the redesign case, two possibilities are taken into account: adaptive and parametric. The adaptive redesign case also begins with the development of physical solutions where it is likely that an addition or change in functionality will occur. The parametric redesign can begin with the development of the architecture workframe based on the existing form of the current design.

This chapter validates the above method by incrementally considering the capability of method constituents and method performance in the context of the validation square technique. Both an original design project and an experimental case study test the method in real situations. The original design demonstrates that the method facilitates the development and exploration of multiple concept layouts. The experiment involving a comparison of the proposed method with the Ulrich and Eppinger (2000) approach shows statistically significant advantages over this approach in terms of the quantity of solutions generated as well as the efficiency with which they are generated.

Chapter 6 – Conclusions and Future Work

This chapter presents the dissertation results as they relate to the hypothesis and to design theory in general. Contributions are stated in terms of how this work helps both support the hypothesis and explain questions about architecture design.

6.1 SUMMARY

Architecture design is complicated largely because there is a discontinuity between function and form. This is a problem because as form is addressed, the designer encounters a sharp rise in the breadth of the design space. In contrast to the simplicity of a functional description, the specification of geometry and material, even in terms of a rough layout, involves a large number of coupled parameters such as the degree of modularity, number of parts, interfaces, manufacturing source, product family constraints, etc. A consequence of this problem is an inefficiency on the part of the designer to effectively proceed from function to form. This difficulty is analogous to the state of affairs in the 1920's and before when the entire design process presented a discontinuity between the customer needs and the final product. Beginning in earnest about that time and ever since, design has generally improved toward a more systematic operation with the introduction, development, and use of design methods.

This research addresses the architecture design problem with the hypothesis that a method based on 1) a formal representation of product architecture and 2) a set of guidelines can lead the designer to architecture design solutions more efficiently in terms of both quantity and quality than conventional design practices. The main objectives of this work are to develop this representation, a set of guidelines, and a process that utilizes these items in the form of a design method. Several basic questions about architecture design are relevant to the completion of these objectives. First, what should be included in an architecture representation? Secondly, how should the information be presented in terms of format? This work utilizes a mental model concept to serve as a framework for the representation. A lexicon of product architecture is developed to provide an explicit set of terms that are relevant to architecture design. A six element notation is used to instantiate the lexicon in a manner that facilitates designer interaction for observing and controlling the design. In terms of guidelines, the central question is what action should the designer impose on the design in order to improve the solution? This research develops these guideline actions using product based knowledge extracted from two empirical studies: an architecture parameter study of eighteen existing products and a product evolution study of thirty product evolutions. In addition to the problem of codifying a broad set of product based knowledge, this work also develops guidelines for two relatively specific topics related to architecture: modularity and product flexibility. Finally, how can the representation and guidelines be used in a cohesive process for design, whether original or redesign? This process effectively becomes the design method sought in this work. A three step method is developed for this purpose. In examining these questions, several contributions emerge and are discussed below.

6.2 CONTRIBUTIONS

This work is partitioned into three main elements and for clarity the benefits of the research are discussed in terms of these items.

6.2.1 Representation of Product Architecture

The representation offers direct support for the main claims of this work and in doing so, it helps explain the properties of a successful design representation. Based on the validation experiment discussed in Chapter 5, the method is an improvement over conventional practices in terms of the efficiency with which alternative architecture solutions are obtained. Given that this experiment is restricted to the first three diagrams of the representation notation, it suggests that the composition of these three diagrams attributes to representation effectiveness. Specifically, the representation is a useful contribution because it provides knowledge of the set of items relevant to architecture design: the lexicon, and knowledge of the format for this information: the notation.

Probably the most characteristic property of the representation is the incremental level of detail among the six diagrams. This allows the designer to take smaller steps toward a solution by beginning with known parameters, such as the external flows, and leading toward more detail without overwhelming the designer in any given step. This demonstration of incremental solution development is another useful contribution and it is based on the proven technique of divide and conquer. Given the nature of embodiment design, which shares a similar nonlinear process, this work suggests that improvements in embodiment design might also be achieved by the development of such a representation analogous to the architecture representation in this work. In addition, this work indicates that a construct such as an 'architecture domain,' defined by the representation, is a useful approach. Therefore this work provides a foundation for establishing such a domain in the overall scheme of the design process. In systems engineering work in particular, where architecture design and embodiment design are currently mixed together as an overall task, inclusion of this domain as a distinct design phase can offer the benefit of an additional verification step prior to embodiment design. This again is more consistent with an incremental approach which is generally effective. One important issue is how granular the design process should become. In the case of architecture design, it appears than the finer resolution afforded by the proposed representation is beneficial. However, at some point these benefits are expected to diminish and it is not clear just when this will occur.

In addition to the overall concept of the proposed representation, another contribution is specifically the Function Layout Diagram (FLD). The FLD offers a means to visualize the spatial locations allocated to each function of a product. This allows designers to accomplish several goals. First, the FLD has the capability of illustrating function sharing with respect to spatial regions. This property is very similar to the wirk element scheme (Jensen, 2000). Secondly, the FLD provides a mechanism for identifying functionally independent regions in space which can serve as an extension for current modular identification techniques such as Stone's three modular heuristics (Stone, 1997) and platform identification techniques by Zamirowski and Otto (1999). The FLD can therefore serve strongly as a tool for partitioning a device into modules and components. Third, the FLD leads to a reasonable format of device decomposition where the designer can evaluate the impact of future unknown changes. This specifically can be useful for the purpose of designing for flexibility.

6.2.2 Guidelines for Architecture Design

The fundamental contribution of the guidelines is a set of product based knowledge that provides direction to the designer. One portion of this knowledge shows the relation of several architecture parameters such as modularity and interfaces to assembly cost. This data is useful in understanding how one can design while taking these factors into account. The powerful benefit from the approach taken in this work is that such data and their related recommendations are given in terms of physical characteristics of the device. A second portion of architecture design knowledge, derived from a separate study of product evolutions, is useful because it captures information about a wider spectrum of architecture design factors as indicated by guideline titles in Table 5.4. Most importantly, the guidelines from this evolution study appear to be quite applicable based on a validation effort that tested their utility with respect to thirty existing products. Not only does this validation suggest that the guidelines have been used to evolve the sample products to their current state, but the data indicates that the guidelines could likely be applied to the existing devices to evolve them even further. The guidelines derived from this study also include product based knowledge of two particular aspects of product architecture: modularity and product flexibility. Due to the importance of modularity and device partitioning in general, one design action within the Partition guideline specifies a new approach for identifying modules based on the FLD. This knowledge extends the current set of heuristics for identifying candidate modules and therefore provides good support for the important problem of designing for modularity. In addition, the Flexibility guideline specifies six ways in which a product can be made more flexible. This flexibility knowledge contributes to those areas of research and practice involving any case where a product will likely change in the future. Benchmarking, redesign, new product development, future product planning, and product portfolio development are current applications that can directly benefit from this new knowledge.

6.2.3 A Method for Architecture Design

The main contribution of this work is a design method which puts the representation and guidelines to use in a process that can be applied to both original and redesign problems. Based on the validation experiments, the method effectively improves the current situation with regard to overall designer efficiency as illustrated in Figure 6.1. Because the process of developing initial layouts becomes more systematic with this technique, it offers a perspective of how less experienced students such as freshman can be introduced to design. Whether engineering freshman or experienced designers, the most important point is that the design method can be taught. Based on the validation of the method, this work presents a new set of reasonably effective design aids for tackling the architecture design task.

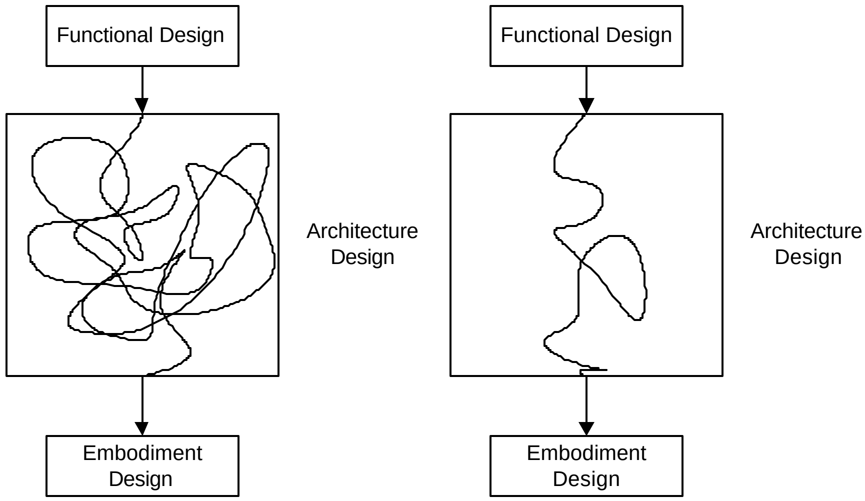


Figure 6.1 The primary objective – a more direct path from function to form

6.3 FUTURE WORK

This current work extends existing areas of research as indicated from the contributions above. Although the objectives for this work are now met, several problems remain and some new problems have emerged. Speaking broadly, a global knowledge plateau for both the theory of architecture design and the applications related to architecture design still seems far away. The following discussion addresses direct extensions of the current work and highlights on going problems that will likely require substantial further effort. A few advanced topics are also proposed that seem promising yet very underdeveloped at this time.

The most direct next steps include further development of the representation to more fully include the effect of designer actions and device operations which might also be classified as the product behavior. This thesis places emphasis on the notion of transforming function to form although one recurring theme in related design research (Benami and Jin, 2002) is the concept of product behavior. The lexicon does include "device operations and user activities" although this term is not represented as well as the other lexical terms in the notation. Presently, this term is best represented by a structure such as an activity diagram used as an input to the architecture representation. It seems reasonable to more fully integrate this concept of product behavior into the representation notation. Perhaps a seventh workframe diagram could serve as the foundation for this additional feature.

Similarly, the problem of portfolio design could be addressed more completely in the representation by possibly reworking the product family diagram so that other complete concept variants within the portfolio are more explicitly included in the notation. Portfolio design and product architecture design are part of the same problem in general. In order to more effectively unify these bodies of work, another future task is to evaluate the guidelines for product and portfolio design and to identify overlapping concepts such as the concept of modularity. With these commonalities identified, it may be achievable to reconcile any non-common aspects, rework them, and develop a single unified approach to designing both portfolio and product architecture. For example, the techniques for identifying modules can likely serve to support the design of both the product family and the individual product. Not all problems of portfolio and product layout will necessarily be able to be combined, yet the reuse of design techniques for both problems is a desirable goal worth investigating.

The problem of evaluating and designing for flexibility suggests two avenues for further study. First, the current techniques discussed in this work need additional attention to be developed more completely especially in terms of validation. Secondly, the "design for flexibility" indicates that in general the "Design for X" notion is one source for future efforts as they relate to product architecture. For example, why not address multiple "Design for X" issues in the context of product architecture? Several possibilities exist including: design for robustness, design for manufacturing and assembly, design for distribution, design for quality, design for the environment, etc. Some of these topics may not fit as well as others in the design phase of product architecture although each topic is a reasonably good prospect at this point.

Probably one of the more interesting aspects of the current work is the selection of an appropriate design notation. In this work, the notation is implemented in the form of a sketch generally. However, other forms of media offer potential advantages to the current approach. In prototyping work, for example, a designer may develop and explore new concepts through the handling and manipulation of physical media such as foam, wood, metal, etc. Anecdotal evidence suggests that the skills of craftsmanship and tinkering are advantageous to a designer's capability. It seems reasonable then to pursue a study to determine how a designer might systematically utilize physical media as a design notation or as a notation supplement. From an architecture or layout design perspective, one of the likely challenges will be the difficulty in dealing with the inherent lower level of abstraction that generally occurs when physically constructing anything. Sketches, symbols, data structures, and other such abstractions offer a degree of ambiguity that is helpful in stimulating creativity based on the concept of abstraction. It is not clear how to achieve a comparable level of ambiguity in physical media.

The issue of physical media raises the more general question of how formal and informal representations can be used together. The reality is that a designer implements a variety of techniques ranging from back-of-envelop calculations to sketches to computer generated solutions based on mathematical models. With architecture in particular, perhaps there are a selected set of formal and informal techniques that can be combined to form a modified architecture design method. For example, can the current representation be used in conjunction with an informal side process, a mini-prototyping process for example, that explores concepts in the physical domain? An approach such as this could potentially help guide engineers by more efficiently dividing their time between the computer and the prototyping shop.

In terms of on-going work and advanced topics that are likely to remain problems for a long time, one area of interest is the application of an architecture method on a computer platform in order to support automated design. The current representation is reasonably well developed as a manual technique, but it could be extended and developed into a computer tool that acts as a conceptual modeler. See Thompson (2000) for a discussion of this application. This representation was geared toward human design as opposed to automated computer synthesis. Although no fundamental assumptions are made that should preclude automated instantiation of the representation, this avenue was not explored. Based on recent efforts by Campbell (2002) in automated design that incorporate both functional and spatial issues, this work may provide a useful departure point for development of a more comprehensive automated architecture design method that extends beyond traditional CAD, shape reasoning (shape grammars), or catalog design.

Additionally, one application of the representation is the development of a product repository. The basic goal of such a repository is to acquire and store relevant design information from existing and prior designs. This information can then be used for product studies or for methods and tools that require knowledge of prior designs. Although a framework currently exists for capturing design information, the architecture representation in this work provides an opportunity to include additional design data from a layout perspective. The creation of a data structure to incorporate architecture design information is a reasonable next step in the development of such a product repository framework. Both the extension of the representation to a more automated mode of operation and the support for product repository development are good candidate directions for future work.

Appendix A: Observations- A Product Evolution Empirical Study

Pencil sharpeners: <u>Observations</u>: Early models used abrasives and also had a long convoluted energy flow path. Early devices also had relatively substantial mounting / alignment devices for the pencil that were typically not necessary. Early models used a large, heavy, expensive crank instead of small gears – greatly oversized early on. Early models did not have provisions for material waste storage. Early models were very open fame looking. The convoluted flow paths persisted quite a while. Both hand held and table mounted did exist. The abrasives probably became extinct since they require cleaning and abrasive replacement. Most included some style of the era – aesthetically. Cutting tools won out relative to abrasive tools probably due to long life. <u>Heuristics</u>: Minimize the length of the energy and material flow path. Perform rough sizing of components to avoid gross over or under sizing. Include a collector or reservoir for waste storage. Make the layout look aesthetically pleasing. Minimize the structure required to control the position of material flow. Close up the framework to have a more continuous product outer surface boundary instead of large holes that reveal the internals of the device. Substitute long lasting parts for replaceable parts.
Post – bolt action Service Rifles: <u>Observations</u>: General reduction in weight. Variations in energy transfer mechanism for operation of action – delayed blowback vs. gas operated. – Gas operated chosen due to fewer complications in case extraction, but this can be dirty depending on the configuration– the G36 seems to have solved this problem in with good collection, restriction, and transport of unwanted by-products. Design moved from integral to modular to support multiple upper variants on a single lower platform. In the case of the Sig 550, the roller recesses were parts that could be replaced. Not so in the HK-G3. In some cases, the designs moved to bullpup design to improve ergonomics. CG location has evolved eg. the G3 and M14 had much weight at the front. Several different manipulation schemes attempted (G3 forward charge handle, AR rear charge handle) Aesthetics is important – perhaps even more so than other “hard” performance metrics such as sight radius. The selection of the M14 over AR10 is one example of this. <u>Heuristics</u>: Reduce overall size and weight. Control unwanted by-products, by collecting, restricting their location, and transporting them to acceptable areas. Use replaceable modules for regions subject to heavy wear. Use a modular design to support multiple variants with a common platform. Place the CG in an ergonomically acceptable location. Generate alternative modes of user interaction in terms of how the user manipulates the device. Do not sacrifice aesthetics for performance.

Toasters: <u>Observations</u>: Early toasters required the user to manually turn the toast in order to heat both sides. The toaster also did not have an auto shut-off. These early toasters required the user to directly place the toast into the heating area. The pop-up toaster design in 1919, still in use today, allowed the user to place the toast into an intermediate location. The toaster would guide and store the toast in the heat area upon use of either human energy input or slow decent by the toaster itself. Once complete, the toaster ejects the toast into a receiving region for the user to accept. <u>Heuristics</u>: Minimize the steps, tools, and time required for the user to operate the device. Create additional material and energy paths if access to the current operating locations is cumbersome. Reroute material and energy paths to be more accessible to the user.
Staplers: <u>Observations</u>: Magazine fed staplers began mostly as open frame style and progressed to an enclosed frame. Use of linkages was simplified to a single lever. Early staplers were over-designed structurally and very heavy. There were two basic layouts that persisted - the linkage style and the simple lever style. The simple lever style is typical today. <u>Heuristics</u>: Reduce the number of linkages. (perhaps an extension of the minimize energy and material flow path) Reduce the size of consumable storage compartments.
Tractors: <u>Observations</u>: The power takeoff (PTO) is a common attribute for tractors once designers realized that the tractor engine can supply energy to auxiliary equipment attached to the tractor such as a shredder. Early tractors had a great amount of void space inside the product region that served little or no purpose. This was reduced over time. <u>Heuristics</u>: Minimize the void space unless the void space is necessary. Branch the energy source to all available sinks that require energy to function.
Writing pens: <u>Observations</u>: progressed from high amounts of user manipulation, dipping in ink, sharpening, etc. to replaceable ink cartridges to low flow rate mechanisms like ball point tips or modern flow-tip ink pens. (Pilot eg.)

<u>Heuristics</u>: Regulate the use of energy or material to the minimum required amounts in order to reduce supply requirements. Use internal storage reservoirs instead of fixed location reservoirs in order to improve mobility.
Chainsaws: <u>Observations</u>: 1947 Oregon – Joseph Cox saw a beetle chewing on wood – he used the idea for a new chain configuration which is still in use today. <u>Heuristics</u>: Apply solutions developed in nature. (biomimetics)
Corkscrews: <u>Observations</u>: Evolved from straight pull, to various forms of mechanical advantage to improve ergonomics. The ratio of output functions or events increases to the number of input functions or events over the evolution of corkscrews. <u>Heuristics</u>: Apply mechanical advantage to reduce the required input forces.
Reloading presses: <u>Observations</u>: The ratio of output functions or events increases to the number of input functions or events over the evolution of reloading presses. <u>Heuristics</u>: Increase the ratio of number of output functions or events to the number or input functions or events. (bang for buck).
Cameras (35mm): <u>Observations</u>: Parts were reduced in general. Pop-up flashes were developed. Large numbers of variants exist – different variants exist for individual functions and different variants exist for functional modules. Cameras were designed to attach to other devices: flashes, tripods, handles. <u>Heuristics</u>: Reduce part count. Make the device collapsible. Generate alternative variants for individual functions. Generate alternatives for individual function modules. Design the device to interface with modular attachments.

Lathes: <u>Observations</u>: The first really modern lathe (~1820's) was the combination of several previous good concepts into one machine. <u>Heuristics</u>: Combine compatible physical solutions to form an alternative concept variant.
Cars: <u>Observations</u>: Sizing of various aspects of cars seems to be a recurring issue: gross over and undersizing is frequent – early tires, lack of sufficient suspension elements to obvious noise variable – rough road, tiny hybrid golf cart cars, compact truck jump seats, <u>Heuristics</u>: Perform rough sizing in order to avoid optimizing around a grossly over or undersized layout.
Pencils: <u>Observations</u>: Progressed from disposable to a form which allows re-supply of material which wears rapidly. Several variations of feed actuation exhibited. <u>Heuristics</u>: Position an orient user interfaces to be ergonomically acceptable.
Ice cream spoons: <u>Observations</u>: Major part reduction, minimization of material and energy flow path <u>Heuristics</u>:
Pocket knives: <u>Observations</u>: Location of interface for opening can provide a major difference in opening time. <u>Heuristics</u>:
Shaving razors: <u>Observations</u>: Began as a device which had very little safety margin – straight razor. This device required resharpening. Disposable razors became common. Motorized shavers are available for some degree of automation and no requirement for shaving cream or water. <u>Heuristics</u>: Reduce supply requirements by eliminating them and substituting a more convenient form

of supply. Provide safety guards.
Coffee makers: <u>Observations</u>: Purification using filters was implemented. The heating effect was also used to transport the water. <u>Heuristics</u>: Purify energy and material flows.
Hole punchers: <u>Observations</u>: User effort drove the evolution of this device. – Minimize user effort <u>Heuristics</u>: Minimize user effort.
Handguns: <u>Observations</u>: Extensive use of polymer with judicious use of steel inserts. Major reduction in parts. Removal of hammer. Glock use safeties which are relatively internal compared to the 1911. <u>Heuristics</u>: Use material variations, surface treatment, or inserts to correspond with high wear and stress areas.
Vacuum cleaners: <u>Observations</u>: Major reduction in size, and weight, improvement in ergonomics, late models have modular attachments. <u>Heuristics</u>:
Bicycles: <u>Observations</u>: Mechanical advantage implemented, suspension implemented, still very expensive. <u>Heuristics</u>:
Key turning device: <u>Observations</u>: Major reduction in weight, simplified energy transfer, collapsible, mechanical advantage for ease of use.

Heuristics:
Shoes: Observations: Progressively improved fit with foot – early models were symmetric. Suspension with soft soles Heuristics:
Radios: Observations: Reduction in size, portable, multiple attachment points – arm, belt, pocket, etc. Heuristics:
Flashlights: Observations: Reduction in size, improved seals, more portable, attachable, ergonomic to fit multiple holding configurations. Heuristics:
CD holders: Observations: Reduction in parts, more robust, easer to manipulate Heuristics:
Umbrellas: Observations: Early models were collapsible, but not as much as later models. In principle, one can have a one piece umbrella mechanism. Heuristics:

Can openers: <u>Observations</u>: Many different variants exist and persist unlike staplers where the configuration converged to a few layout solutions. Multi-function devices appear often. (can-opener / corkscrew device) <u>Heuristics</u>: Combine multiple devices to make a new device.
Prosthetic legs: <u>Observations</u>: Early models were peg-legs, later models were flexible, later models were adjustable to multiple levels of stiffness and damping, latest models are microprocessor controlled to automate the changes in stiffness and damping. <u>Heuristics</u>:
Telephones: <u>Observations</u>: Generally became smaller and portable. Progressed from mechanical to hard wire electrical to battery powered. <u>Heuristics</u>:
Hair dryers: From Ashby, Michael, F., *Materials Selection In Mechanical Design*, 2nd Ed. Butterworth Heinemann. <u>Observations</u>: Aesthetically driven in layout shape. Major layout shifted from side intake to rear intake. <u>Heuristics</u>: Minimize the number of changes in spatial direction of energy and material flow.

Appendix B: Design Issues Leading to Candidate Lexicon Items

1. Components
2. Modules
3. Coupled functions
4. Energy flows
5. Material flows
6. Signal flows
7. Interfaces
8. Structural components (frame / housing)
9. Manufacturing choices
10. Relative motion
11. Joining / fastening choices
12. Assembly operations
13. Material choices
14. Scale
15. Sizing of device / capacity
16. Existing layout
17. Global layout
18. Local layout
19. Design parameters
20. Performance parameters
21. Noise parameters
22. Specifications
23. BOM references
24. Redundancy of components / functions
25. Degree of need
26. Degree of need satisfaction
27. Component history
28. Functional topology
29. Physical topology
30. Function – form mapping
31. Effectiveness of a product region with respect to satisfying requirements
32. Layout efficiency - Layout relative to ideal
33. Usage of design principles within a region
34. Degree of flexibility / constraint for a region
35. Dependence on a process choice for a region
36. Dependence on a manufacturing / assembly operation for a region
37. Dependence on product family platform
38. Components providing primary / supporting functionality
39. Components providing interface functionality
40. Shared components between modules
41. Shared components between variants
42. Alternative regional solutions
43. Alternative partitions
44. Maintenance issues
45. Accessibility directions
46. Life cycle issues
47. FMEA for regions / parts / modules
48. External interfaces constraints

49. OEM vs. custom fabrication parts
50. 3-D image of parts
51. Aesthetics – Industrial Design
52. Novelty
53. Reliability
54. Robustness
55. Strengths / weaknesses
56. Performance measures for a region / module / component / etc.
57. Weight / Factor of Safety / Accuracy / Precision – “Generic performance measures”
58. Major families of solutions – parents of a tree of related solutions
59. Uncertainty
60. Complexity – both quantity and difficulty
61. Critical path factors
62. Housing / structural frame / base
63. Multiple configurations / change of state
64. Thermal effects
65. Fluid effects
66. Vibration effects
67. Impact effects
68. Disturbance effects
69. Acoustic effects
70. Optical effects, (many physical effects possible)
71. Module partitions / Component partitions
72. Hands-on factors – texture, smell, temperature, compliance, etc.
73. Human factors – style of grip for example
74. Operational states – open / closed
75. Operation based parts
76. Wear / corrosion / fatigue / degredatation
77. Technical difficulty
78. Fit and finish
79. Ease of use / operation
80. Activities / usage
81. Use with accessories
82. Interaction with environment
83. Production of bi-products / waste
84. Ease of cleaning
85. Usage of free-resources (eg. Gravity)
86. Duplication of / or divergence from existing technology
87. Mechatronic systems
88. Control systems
89. Sensors / actuators
90. Effectiveness of a region with respect to a performance metric
91. Packaging
92. Political correctness – image of the device
93. Degree of smoothness
94. Items of major importance
95. Accessibility for difference users
96. Worst case usage scenario
97. Probability of success given a design change
98. Design contraindications given a physical solution choice
99. Ease of optimizing a physical solution

100. Vulnerabilities of a region / module / component / etc.
101. Ease of rework
102. Size of design space in which workable solutions lie
103. Adverse effects related to a physical solution
104. Volume of design
105. Degree of propagation throughout device given a physical solution choice
106. Delayed decision limits – deadlines
107. Benchmarking
108. Variant function structures / optional functions
109. Variant activities and operations
110. Stability / resonance / dynamic issues
111. Cost / benefit factor
112. Driving factors for a physical solution
113. Consistency with good design practice
114. Effective use of standards
115. Safety
116. Fail safe modes
117. Need for calibration
118. Adjustability
119. Market segments
120. Legal liability
121. Physical effects modeling
122. Consequences of the layout
123. Level of completeness
124. Consistency with company resources
125. Recycling issues
126. Sophistication of physical effects involved
127. Location on the Kano diagram
128. Type of prototyping that is most appropriate
129. Type of mathematical modeling that is most appropriate
130. Compliance opportunities
131. Opportunities for separation / merging
132. Opportunities for reduced DOF or increased DOF
133. Opportunities for fewer / greater material variations
134. Opportunities for part reduction
135. Opportunities for complexity reduction
136. Opportunities for interface reduction or interface complexity reduction
137. Opportunities for increasing the intra / inter – interface ratio
138. Opportunities for improvements in general
139. Color
140. Energy storage
141. Allowable component size variations
142. Material gradients
143. Homogeneous / isotropic vs. composite
144. Types of modularity
145. Compliance with legal statutes
146. Surface finish
147. Conflicts among product regions
148. Weather protection
149. Force loading capacity
150. Drop test

151. Child-proof
152. Misuse-proof
153. Pinch points
154. Weak points
155. Patentability
156. Tamper proof
157. Reverse engineering proof
158. Potential spin-offs
159. Modification / Retrofit
160. Use of adhesive
161. Use of lubricant / fluids
162. Opportunities for shrink fit
163. Marketing selling point
164. Wiring conduits / connections
165. Material handling items
166. Insulation
167. Thermal gradients
168. Large motions / small motions – degree of importance
169. Unwanted physical effects
170. Tradeoff between complexity and simplification
171. Price
172. Delivery properties
173. Joints
174. Stress
175. Nested parts
176. Stiffness / damping
177. Pre-inventive concepts / forms
178. Perturbations / morphing history of a physical solution as it evolves
179. Component availability
180. Physical principles applicable
181. Degree of completeness
182. Feasibility
183. Risk
184. Ease of repair
185. Upgradability
186. Product cycle time
187. Undesireable operational by-products
188. Consumables / revenue streams
189. Planned obsolescence
190. S-curve
191. Product evolution / % innovation
192. Supply chain management
193. Accessibility
194. Exportability
195. Part incumbency (sacred cows)
196. Goal of redesign (the new market hook)
197. Retrofit capability
198. Corporate goals
199. Perceived quality
200. Misuse / noise
201. Relation to culture

202. Economic factors
203. Time to market
204. Launch window
205. Geographic location
206. Shelf life
207. Tax implications
208. Stock price / share holder expectations

Appendix C: Function Layout Diagrams for 30 Products

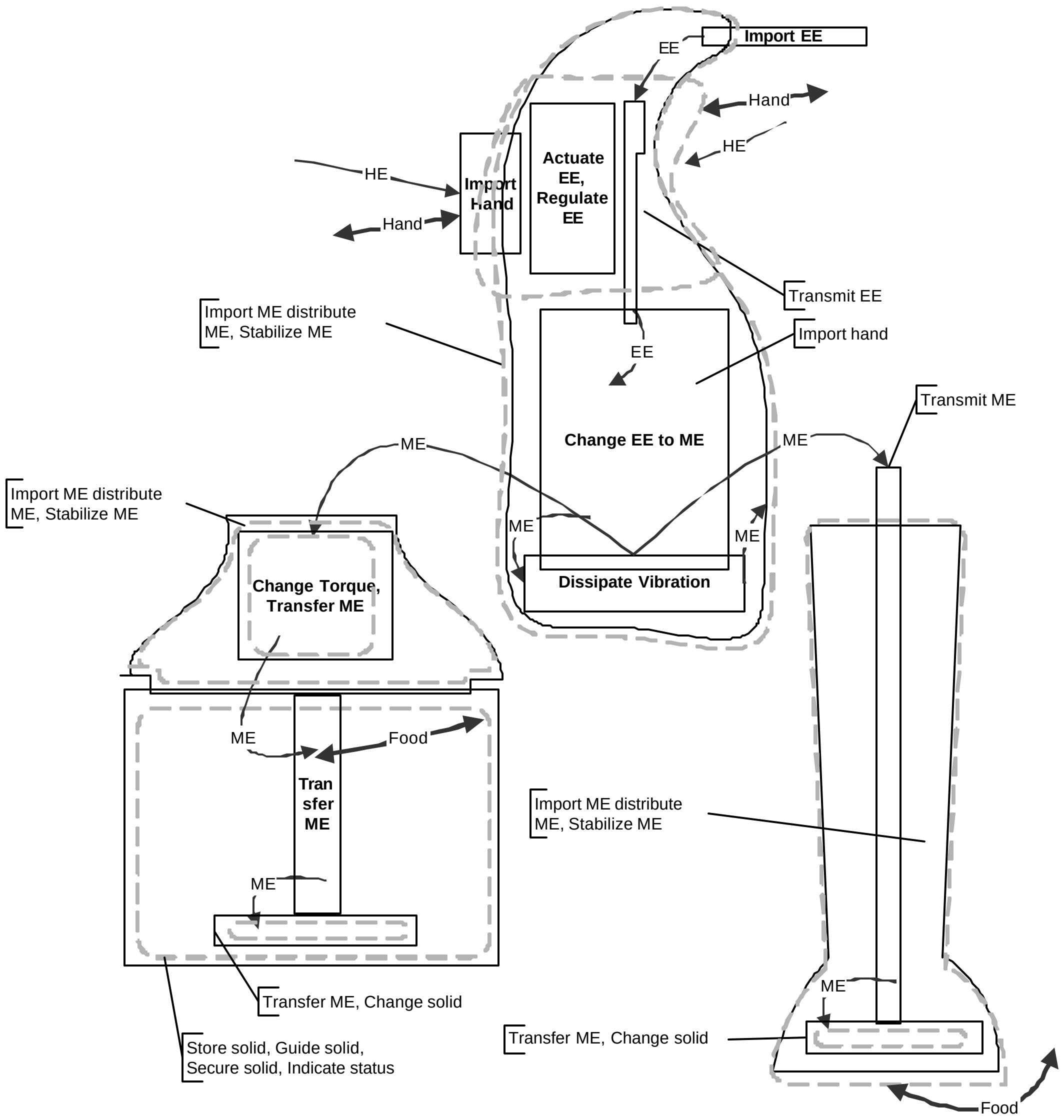


Figure C.1 GE hand blender

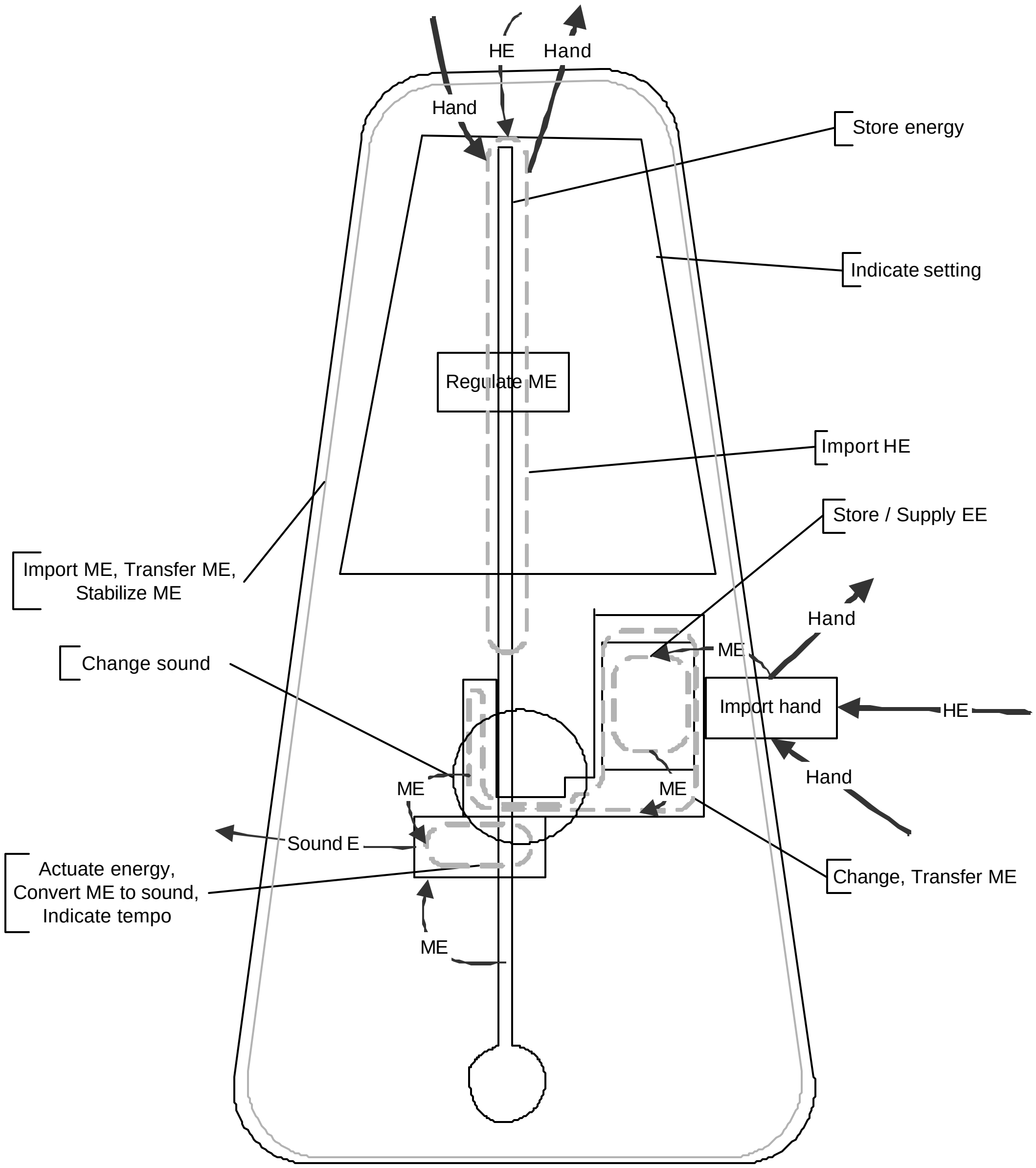


Figure C.2 Metronome

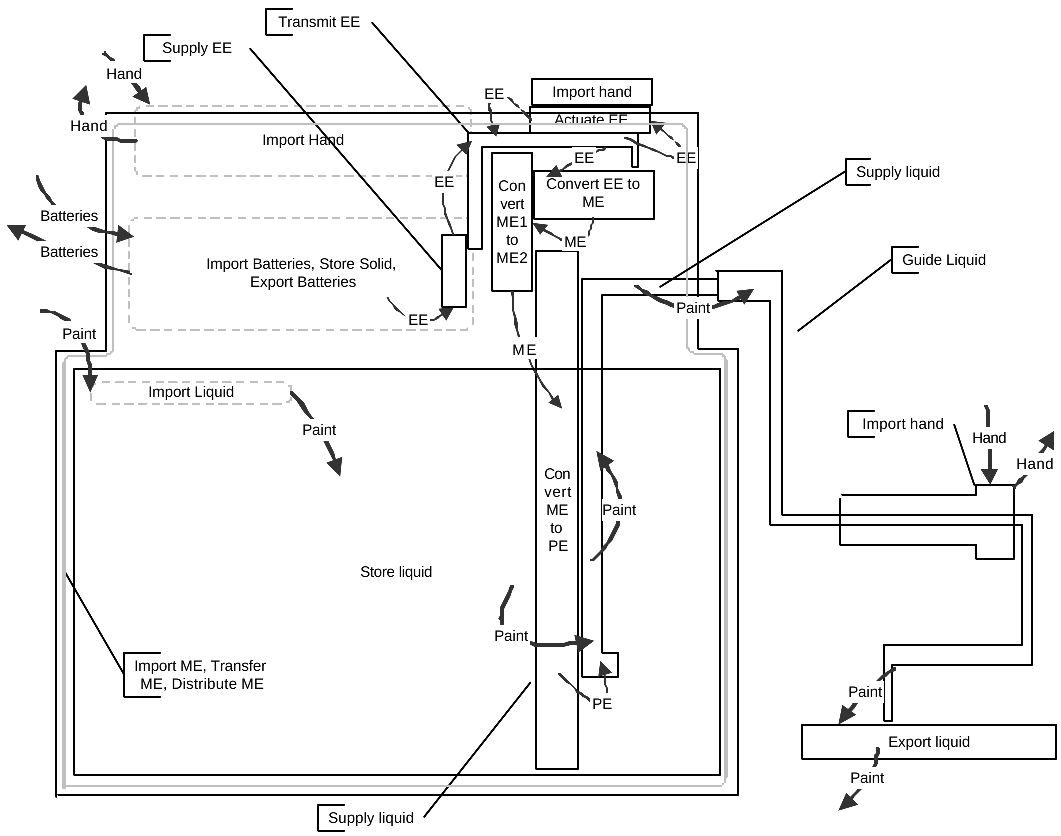


Figure C.3 Wagner paint roller

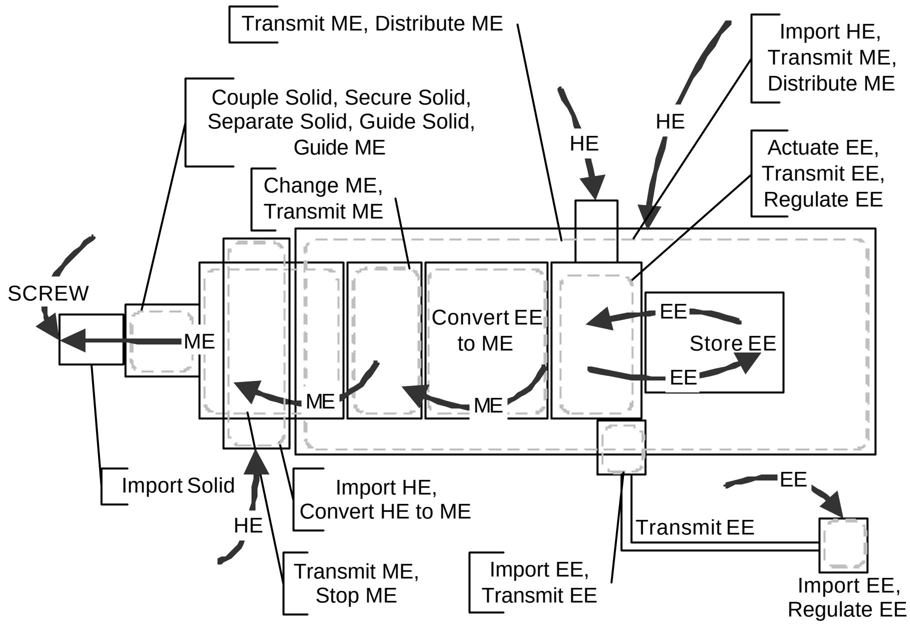


Figure C.4   Skil Twist screwdriver

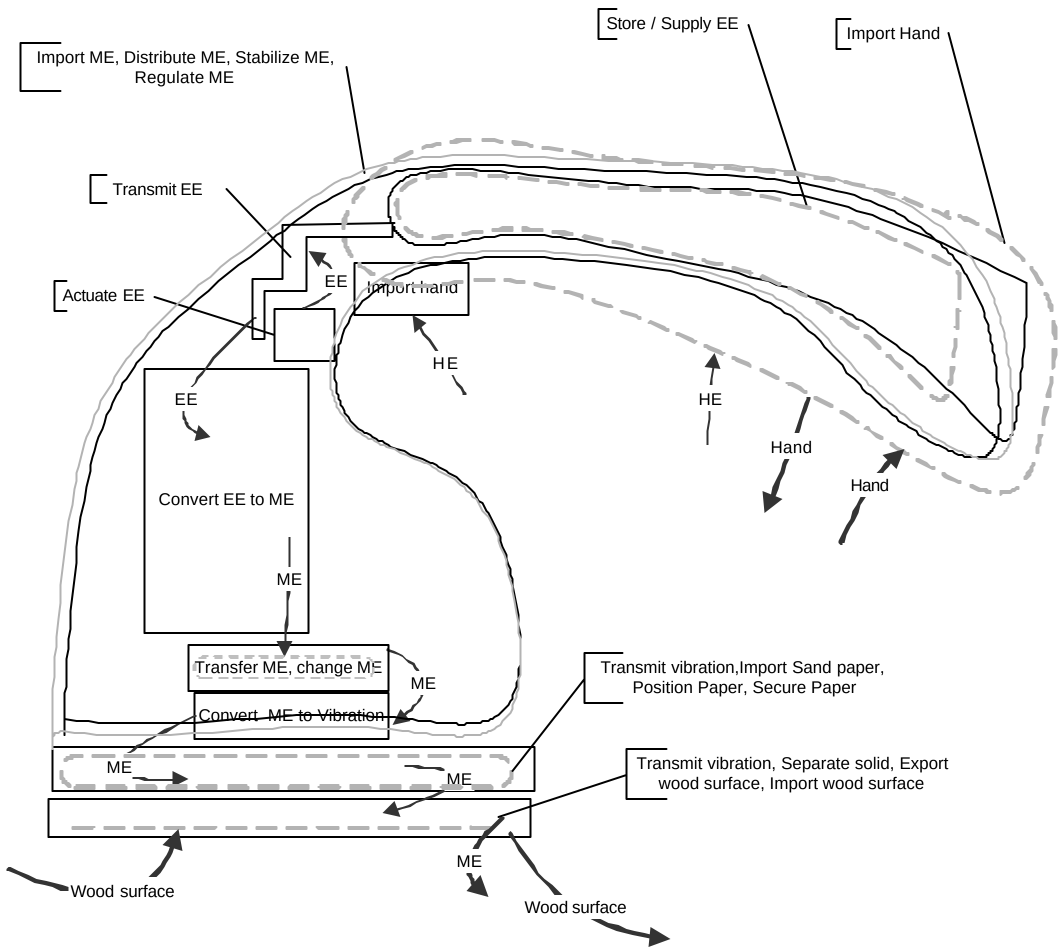


Figure C.5 Freedom cordless hand sander

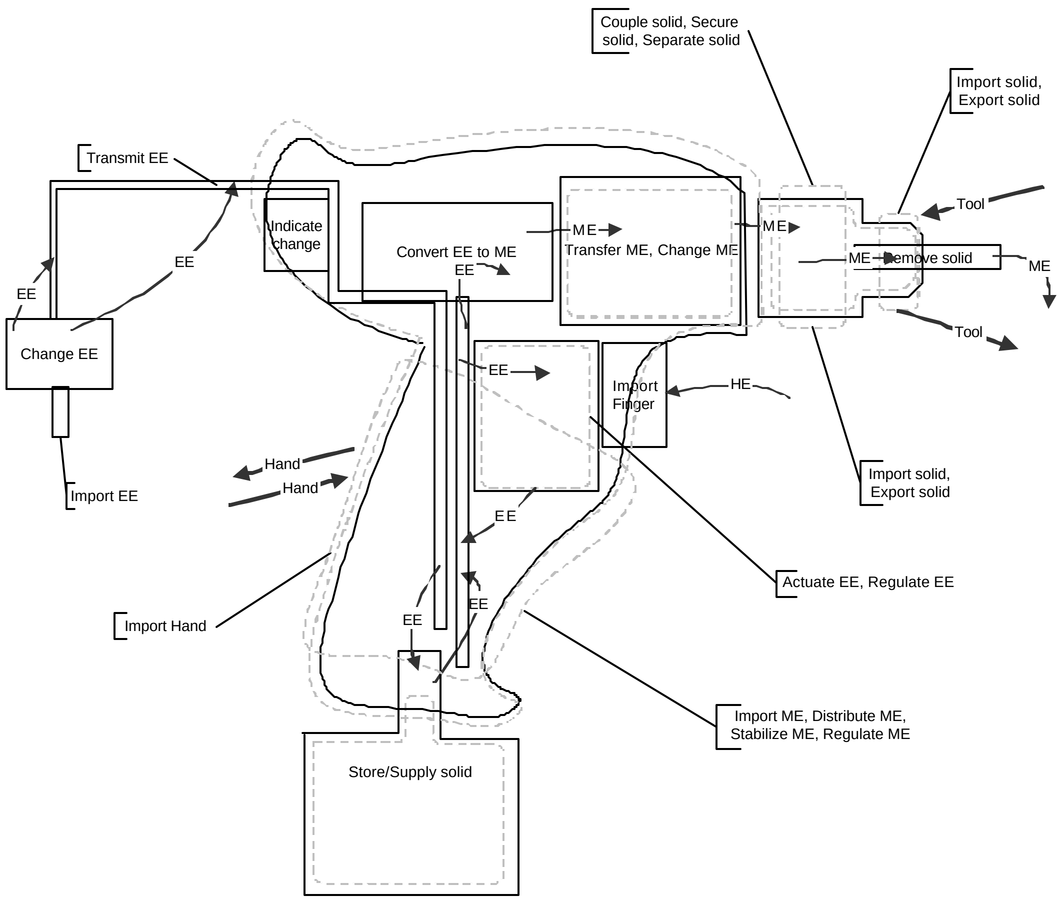


Figure C.6 Handi-Works mini drill

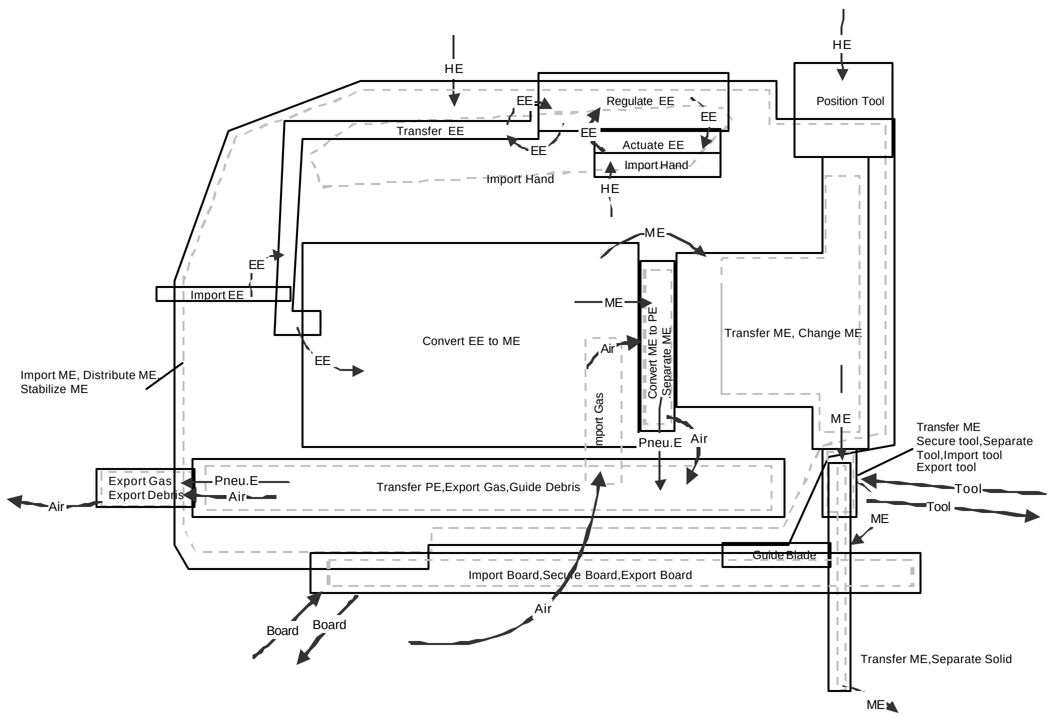


Figure C.7 Black & Decker Jigsaw

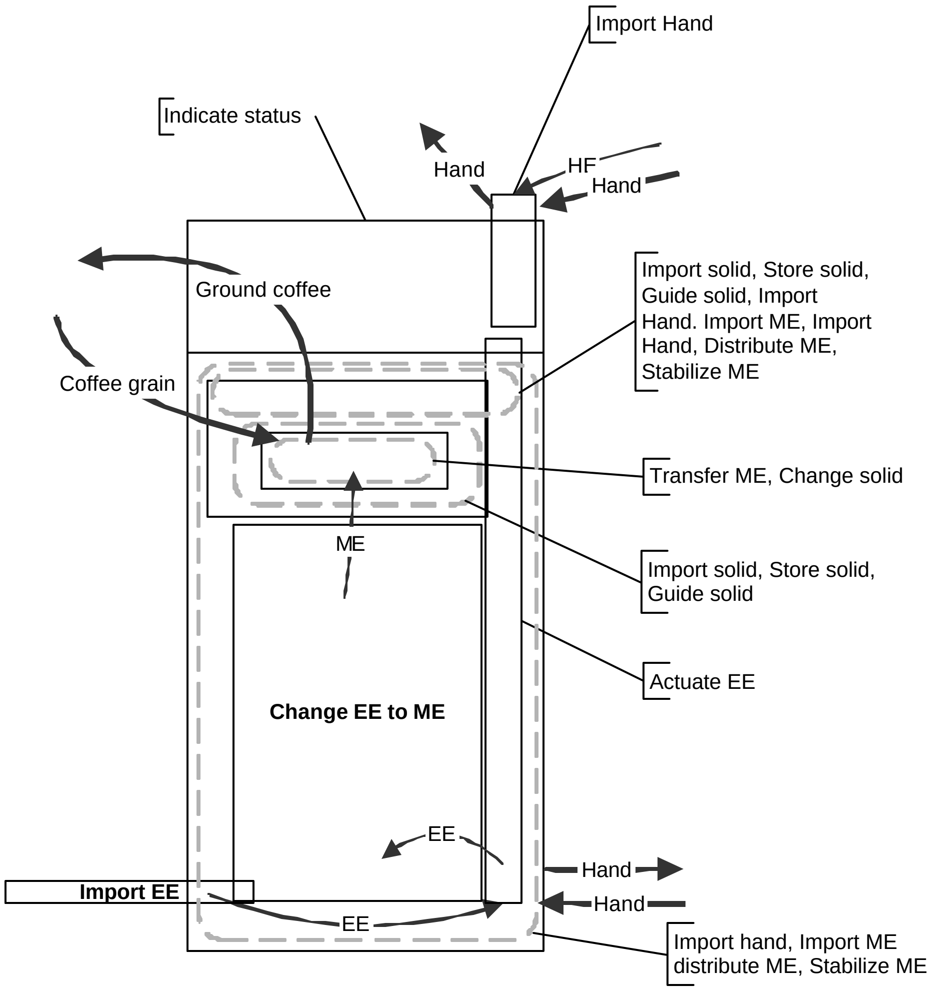


Figure C.8 Braun coffee grinder

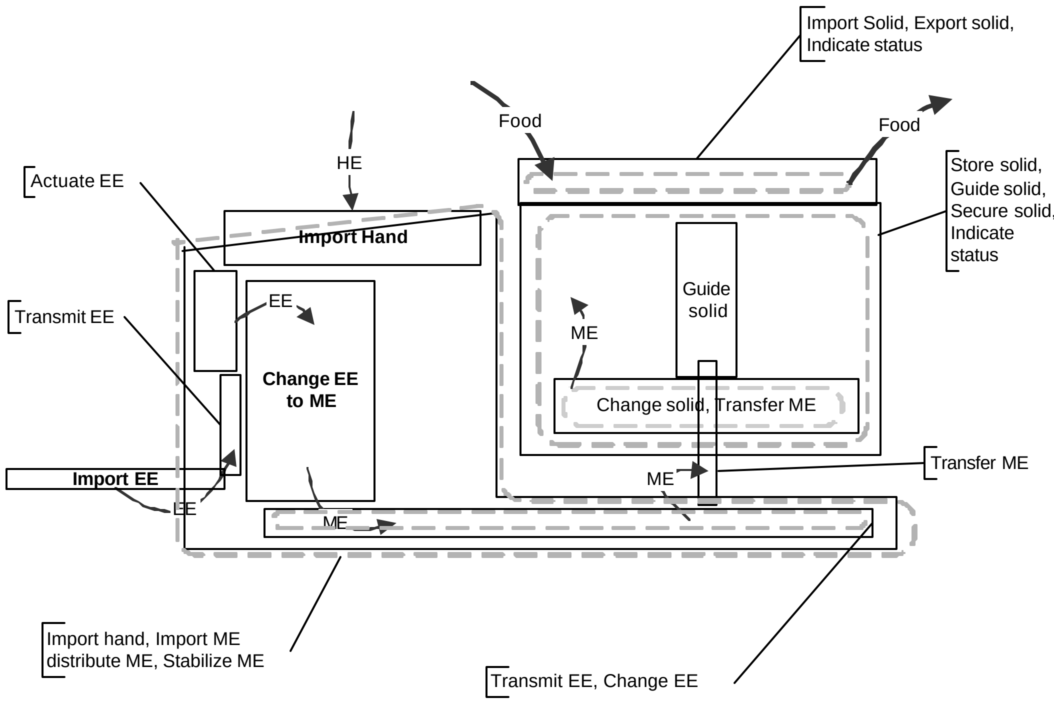


Figure C.9 Black and Decker Handy Chopper

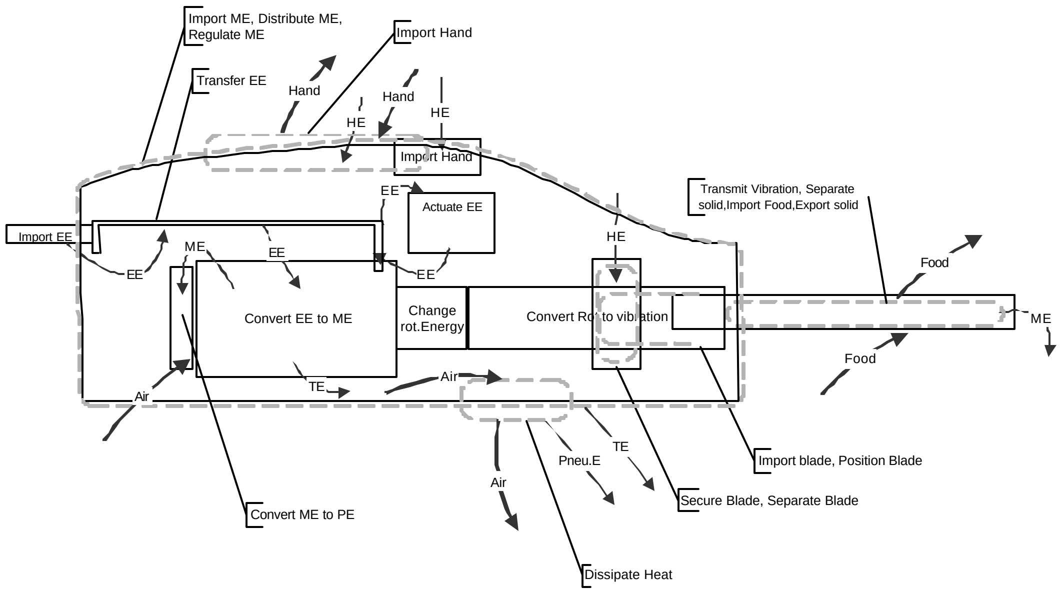


Figure C.10   Toastmaster electric knife

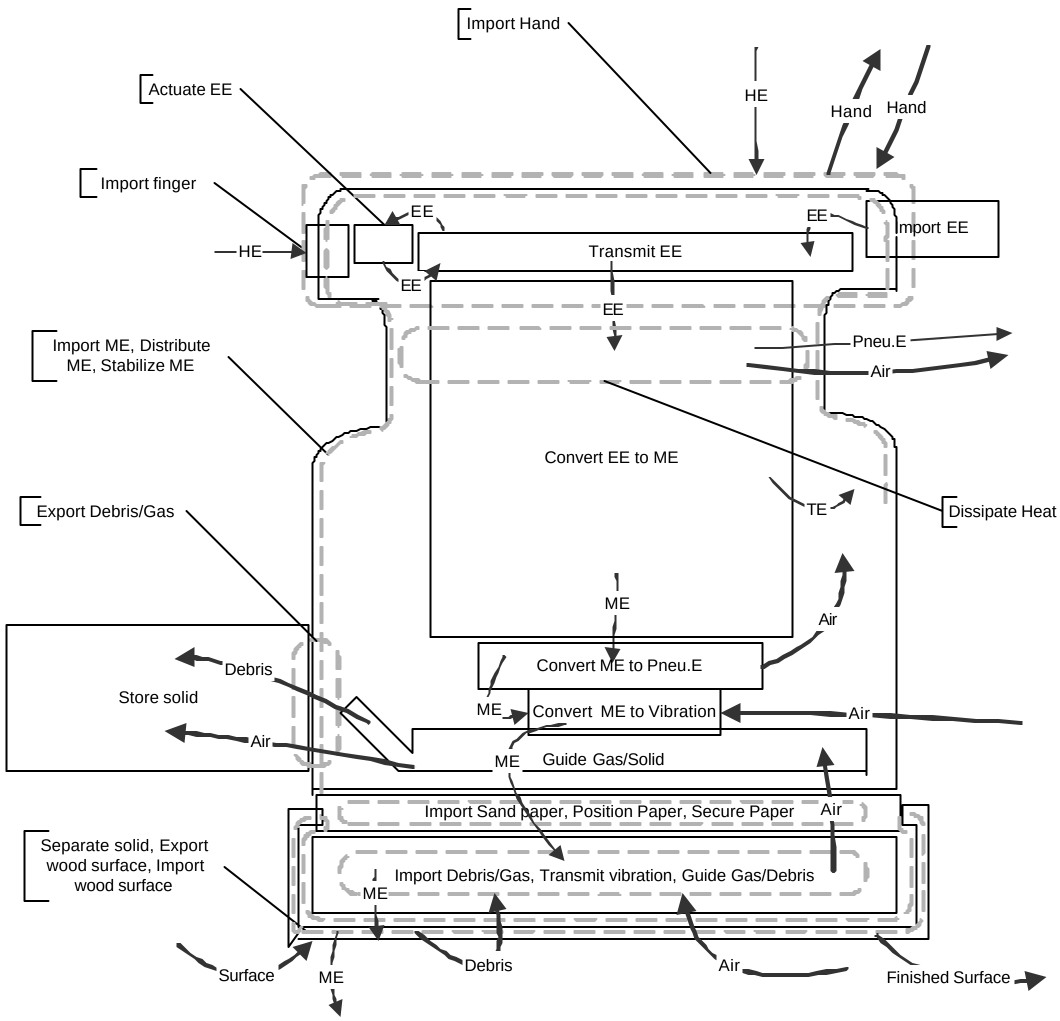


Figure C.11 DeWalt palm sander

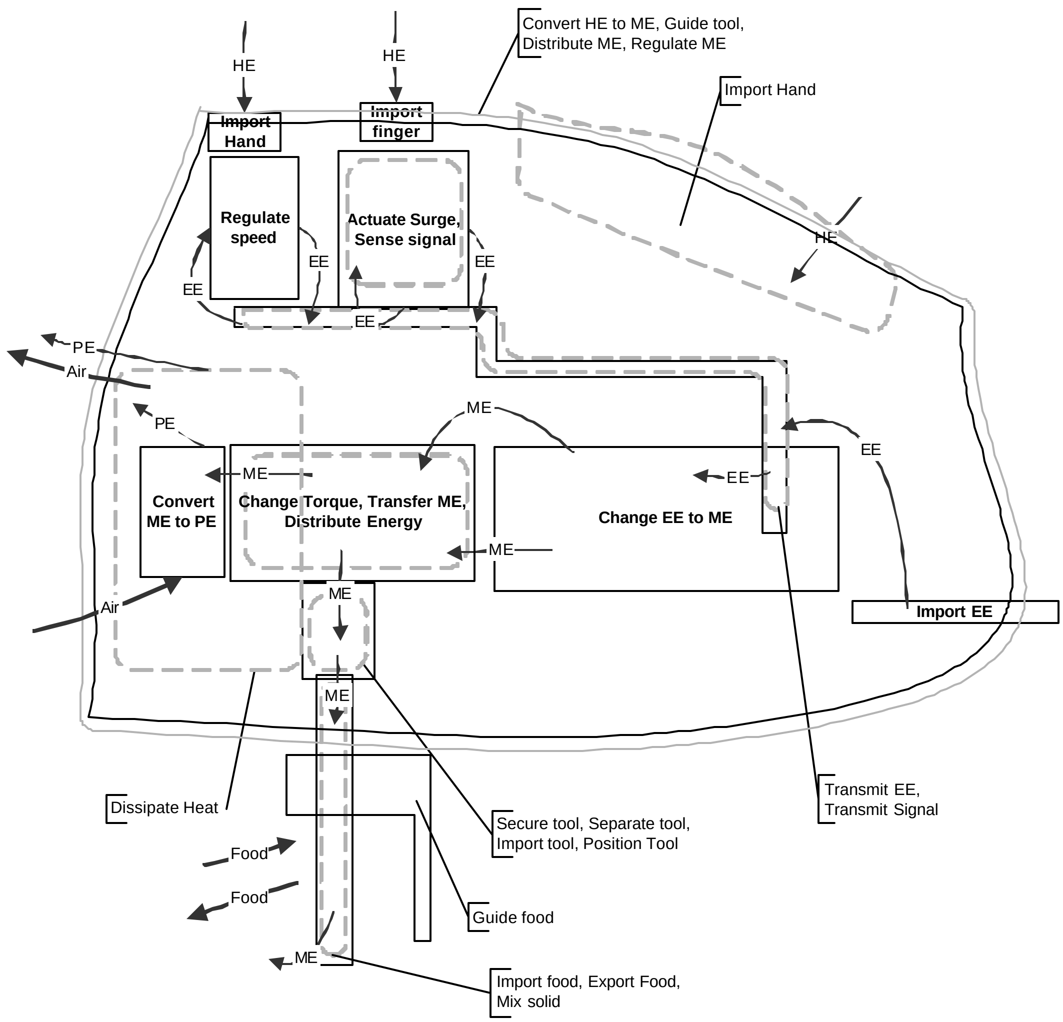


Figure C.12 Black & Decker hand mixer

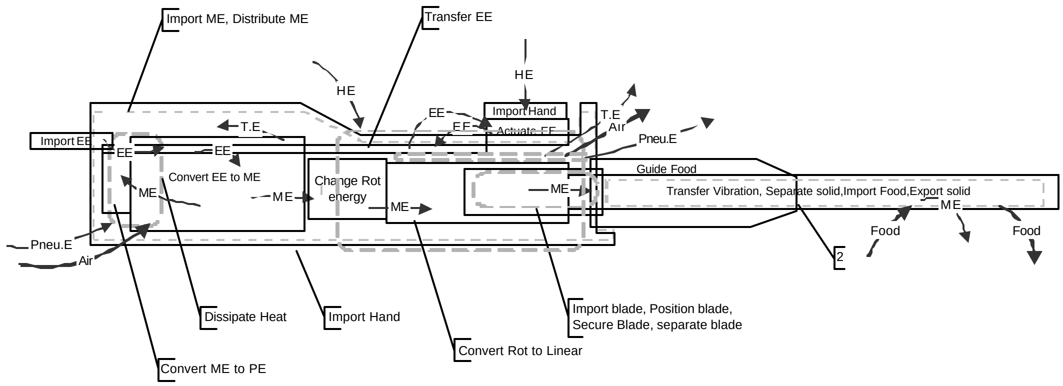


Figure C.13 Black & Decker electric knife

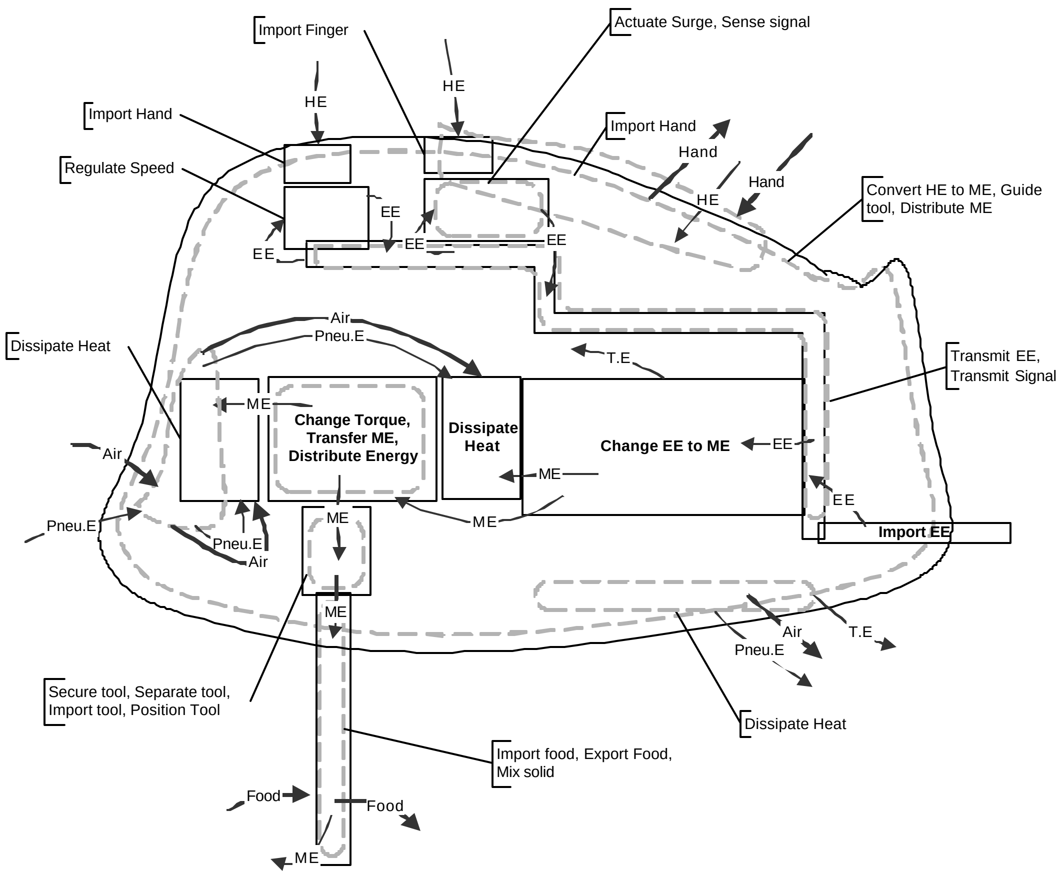


Figure C.14 GE hand mixer

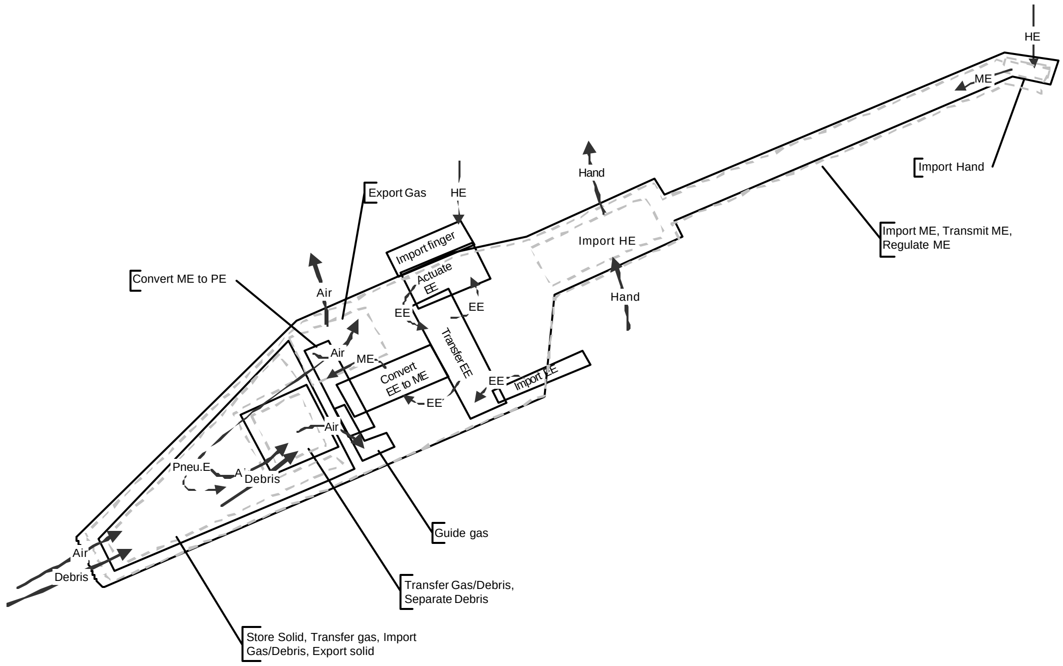


Figure C.15 Bissel hand vacuum

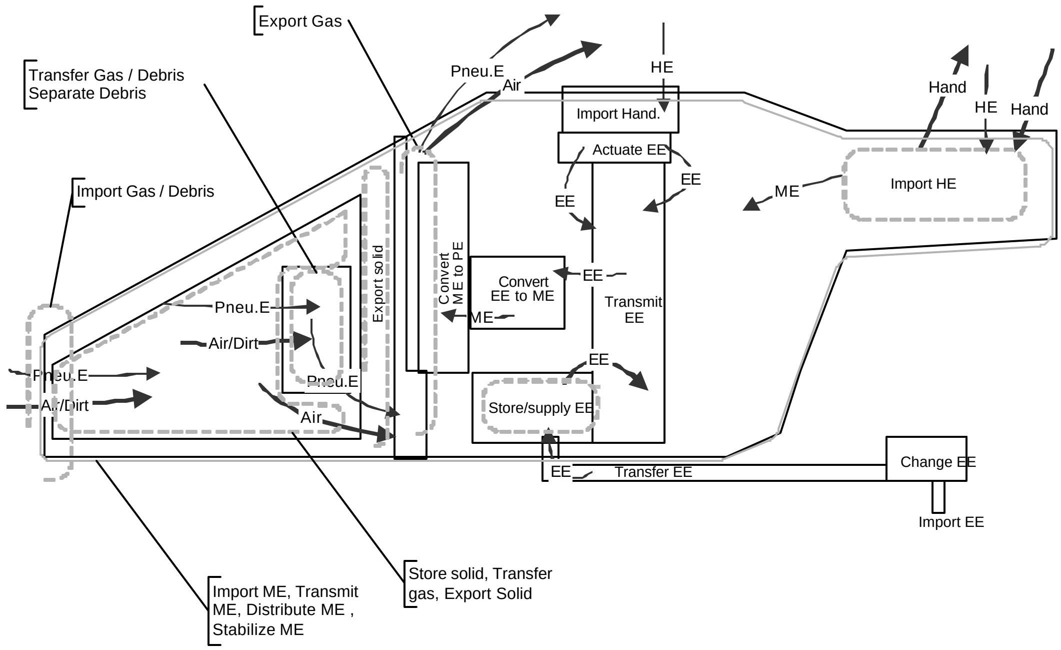


Figure C.16 Dustbuster

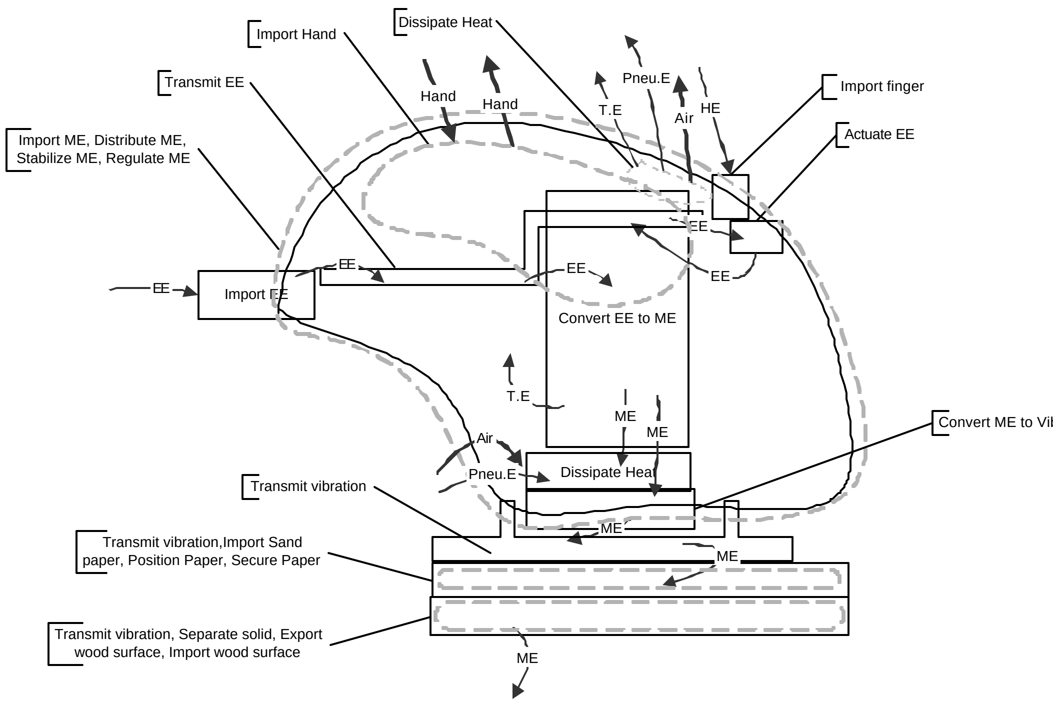


Figure C.17 Black & Decker hand sander

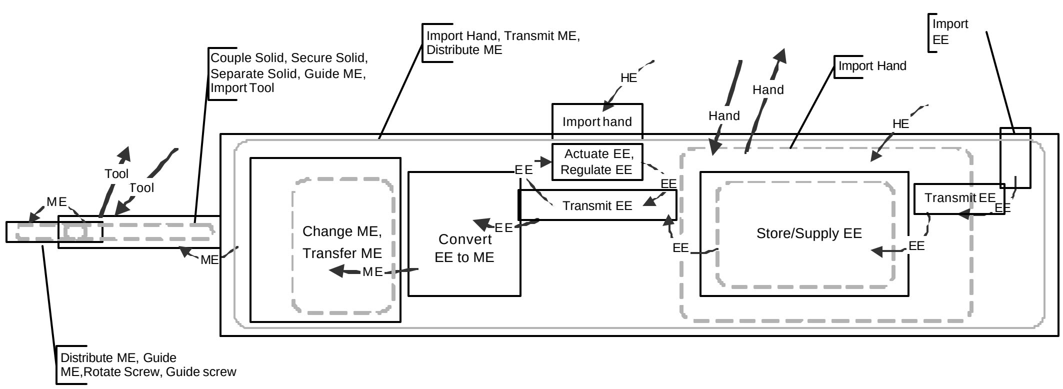


Figure C.18 Handi-Works screwdriver

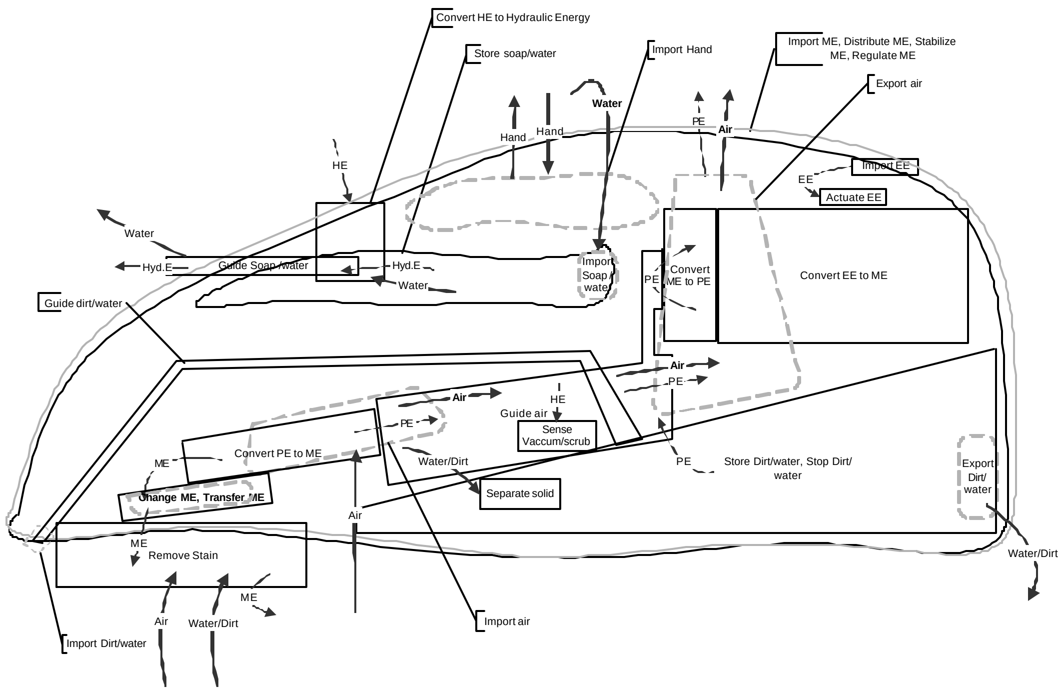


Figure C.19 Dirt Devil Spot Scrubber

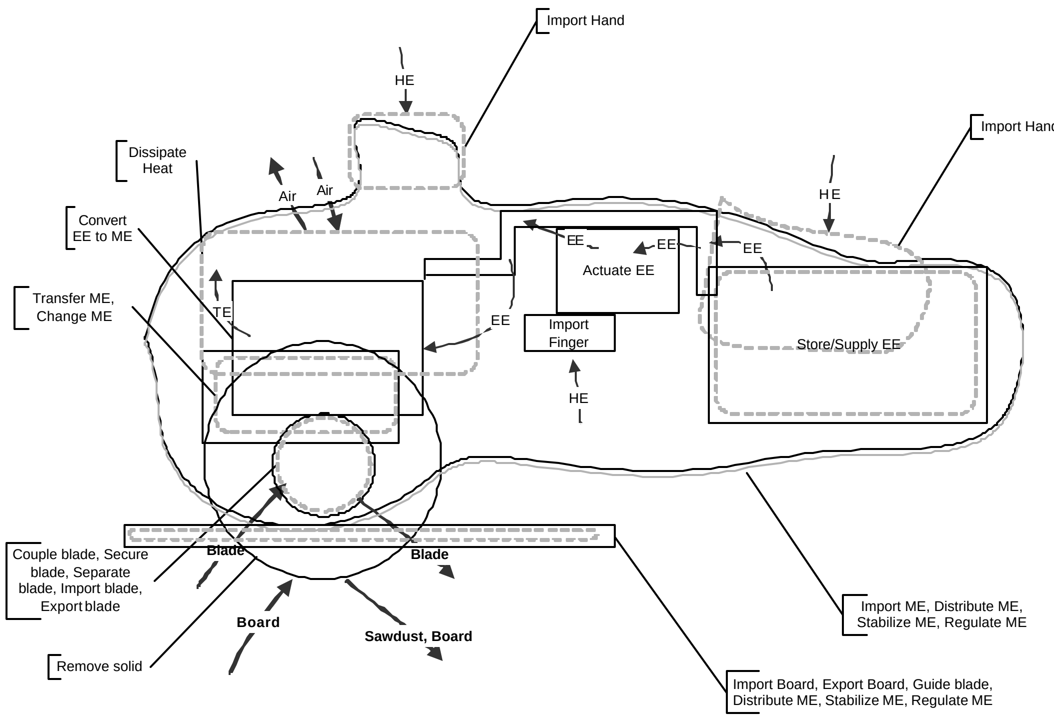


Figure C.20 Versa-Pack Saber Saw

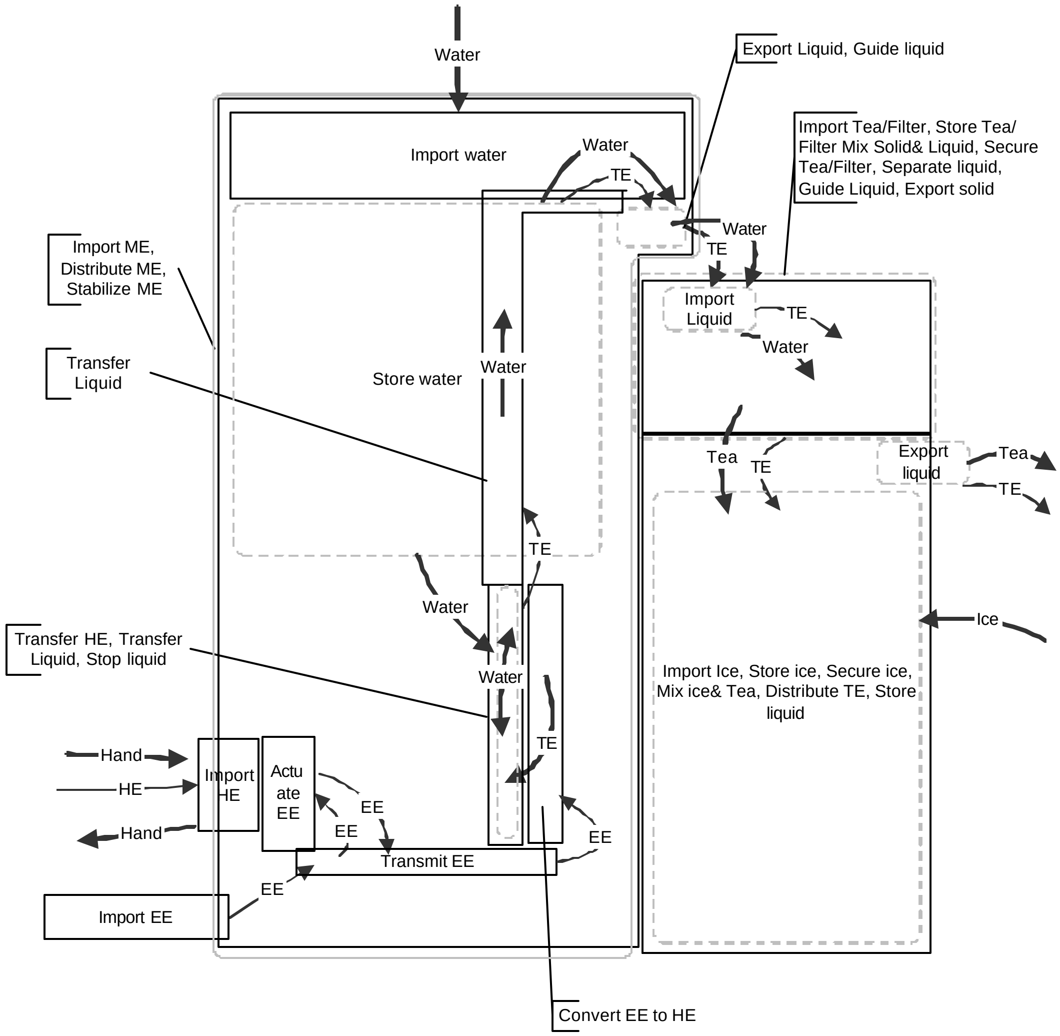


Figure C.21 Mr. Coffee Ice Tea maker

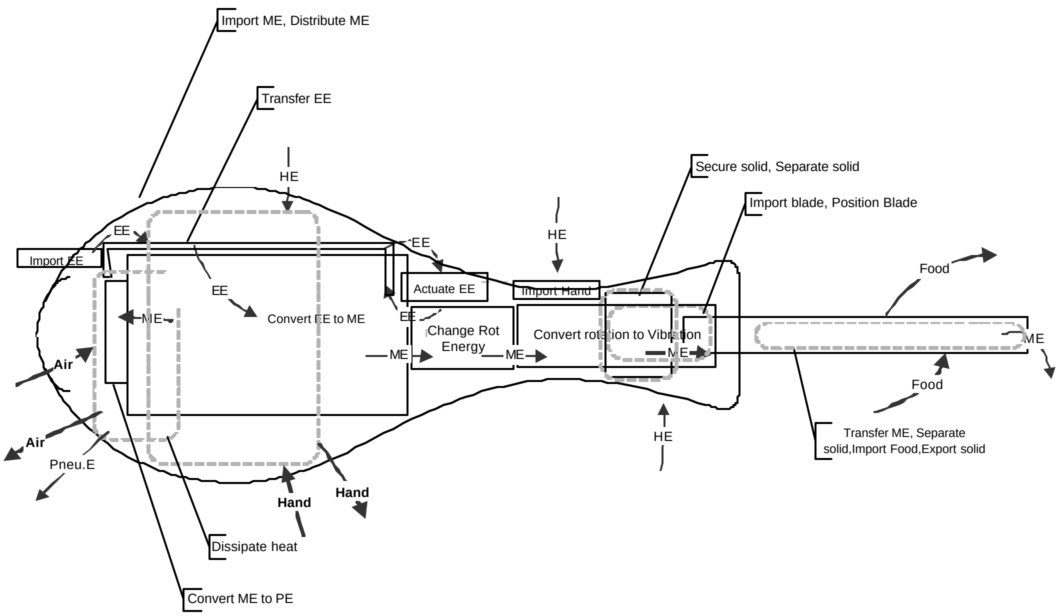


Figure C.22 GE electric knife

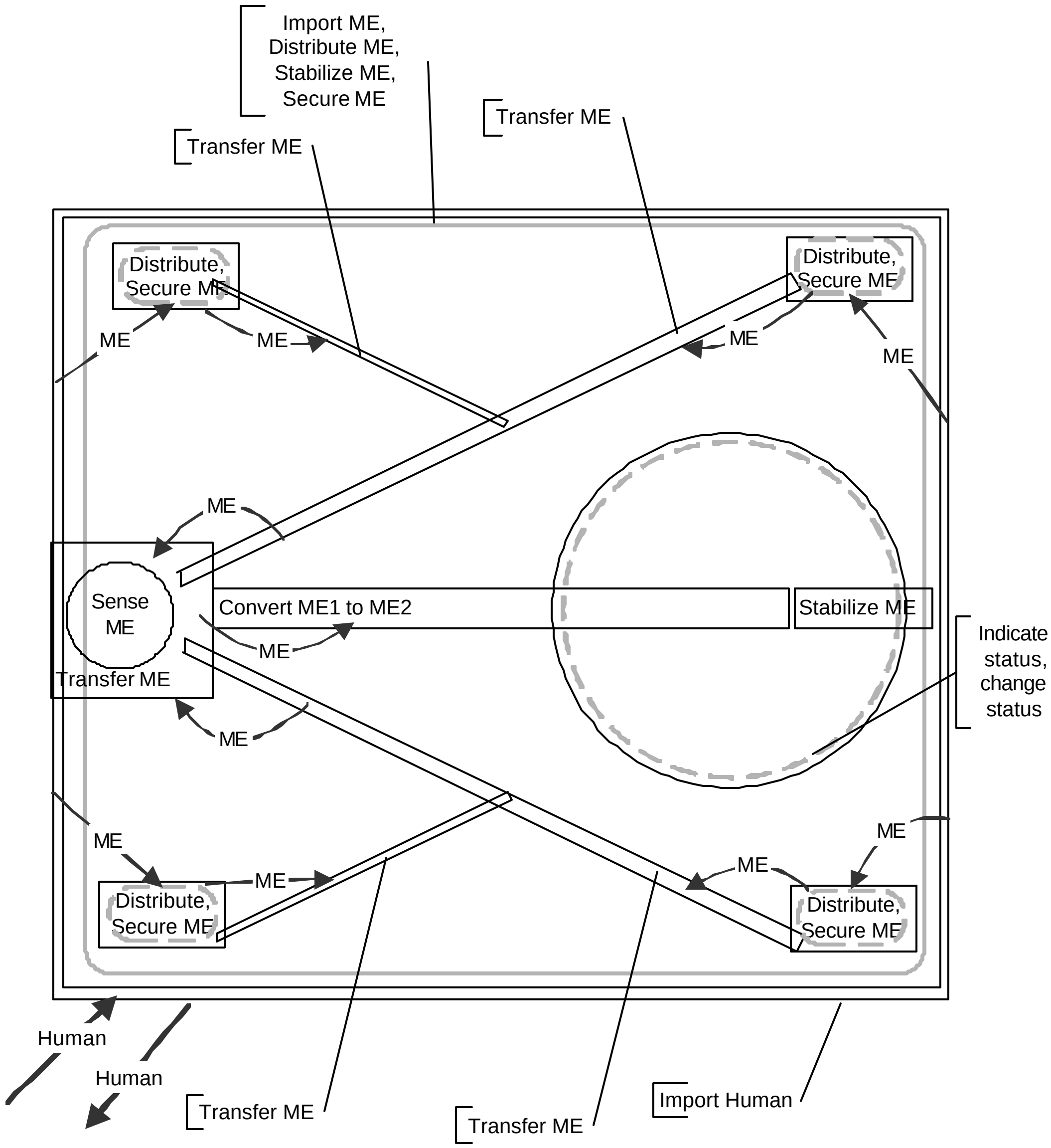


Figure C.23 Metro weighing scale

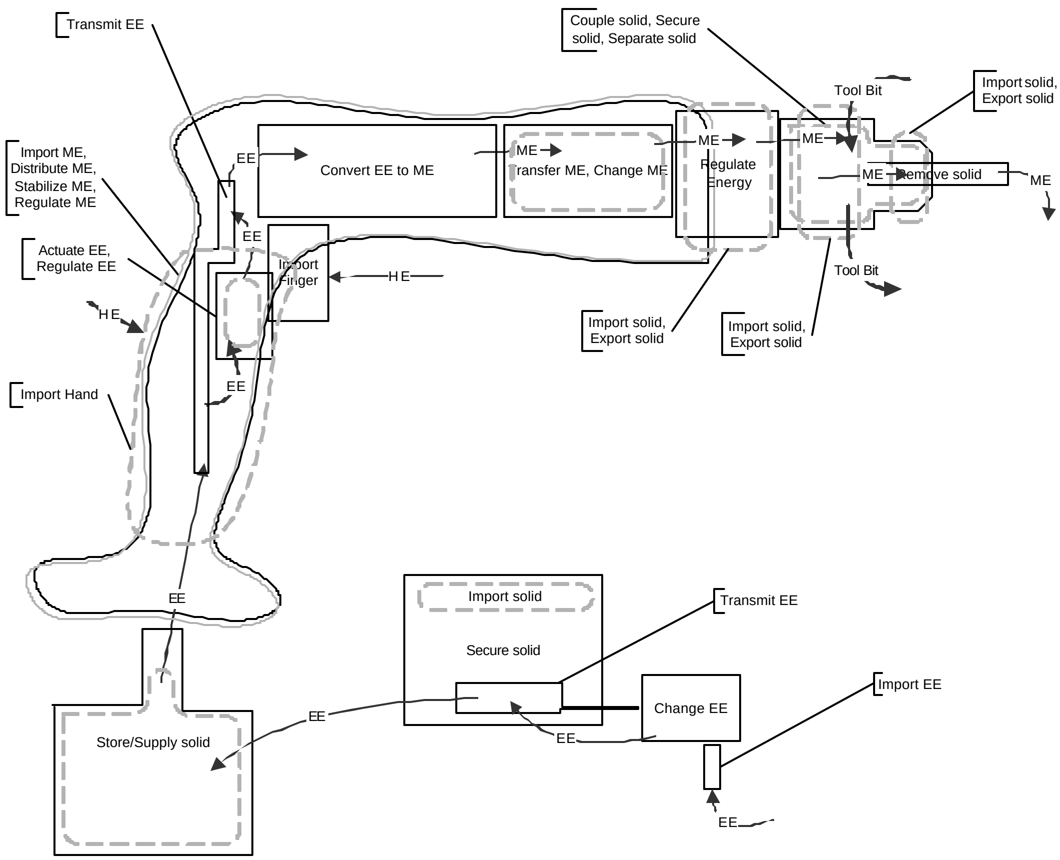


Figure C.24 Black & Decker cordless drill

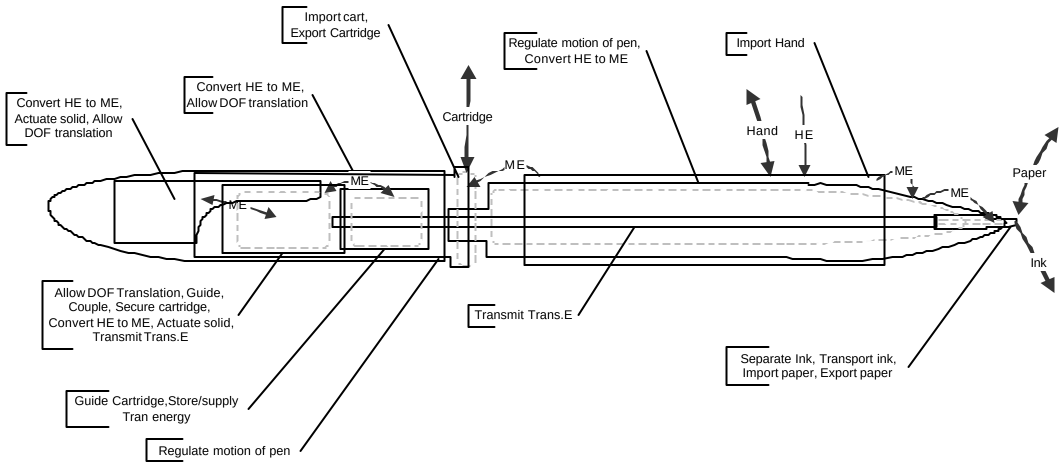


Figure C.25 3-in-one pen

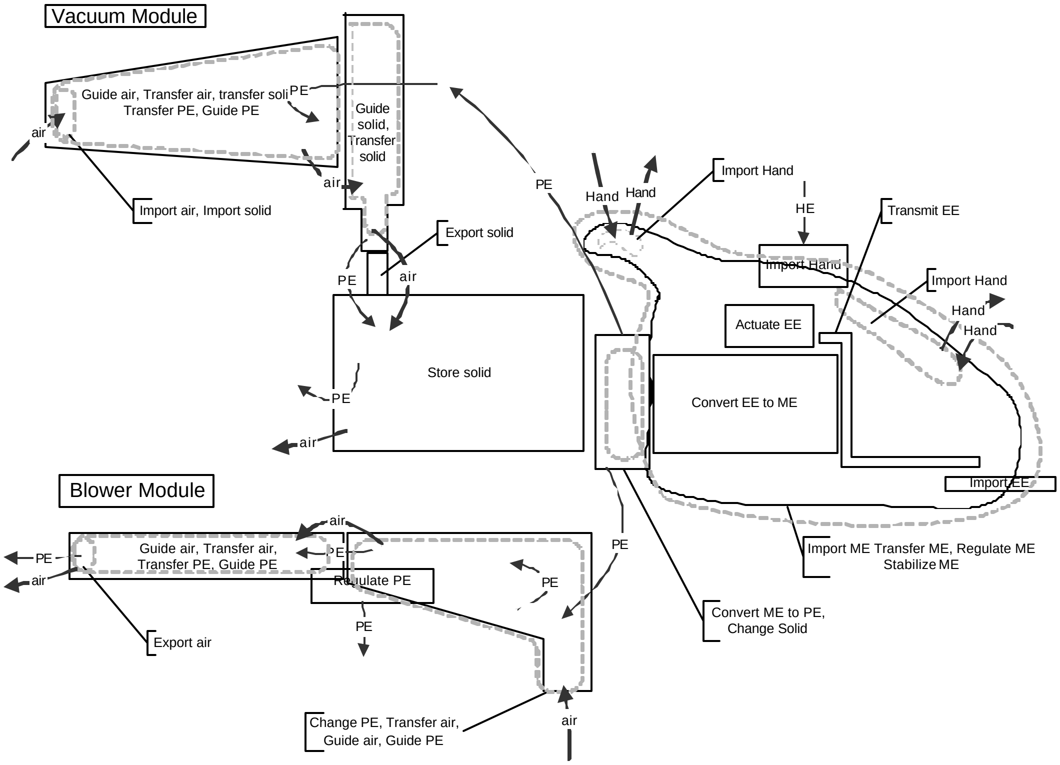


Figure C.26 Black & Decker leaf blower

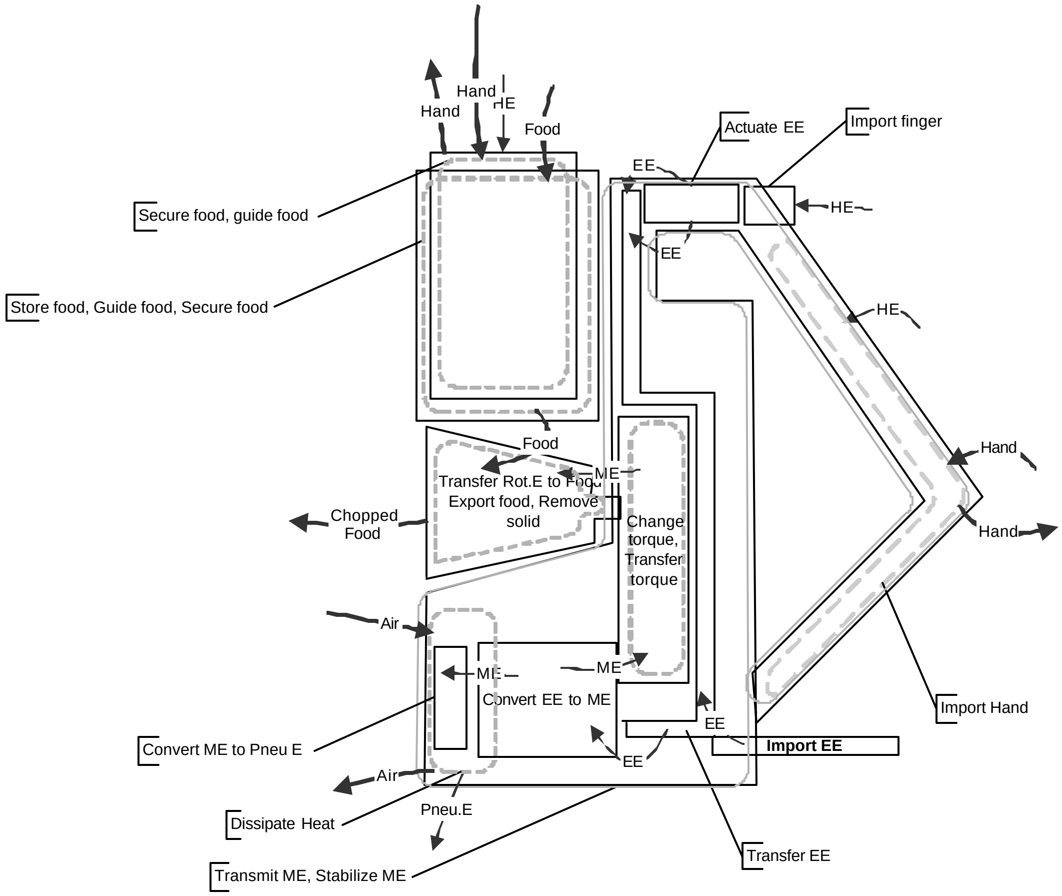


Figure C.27 Presto salad shooter

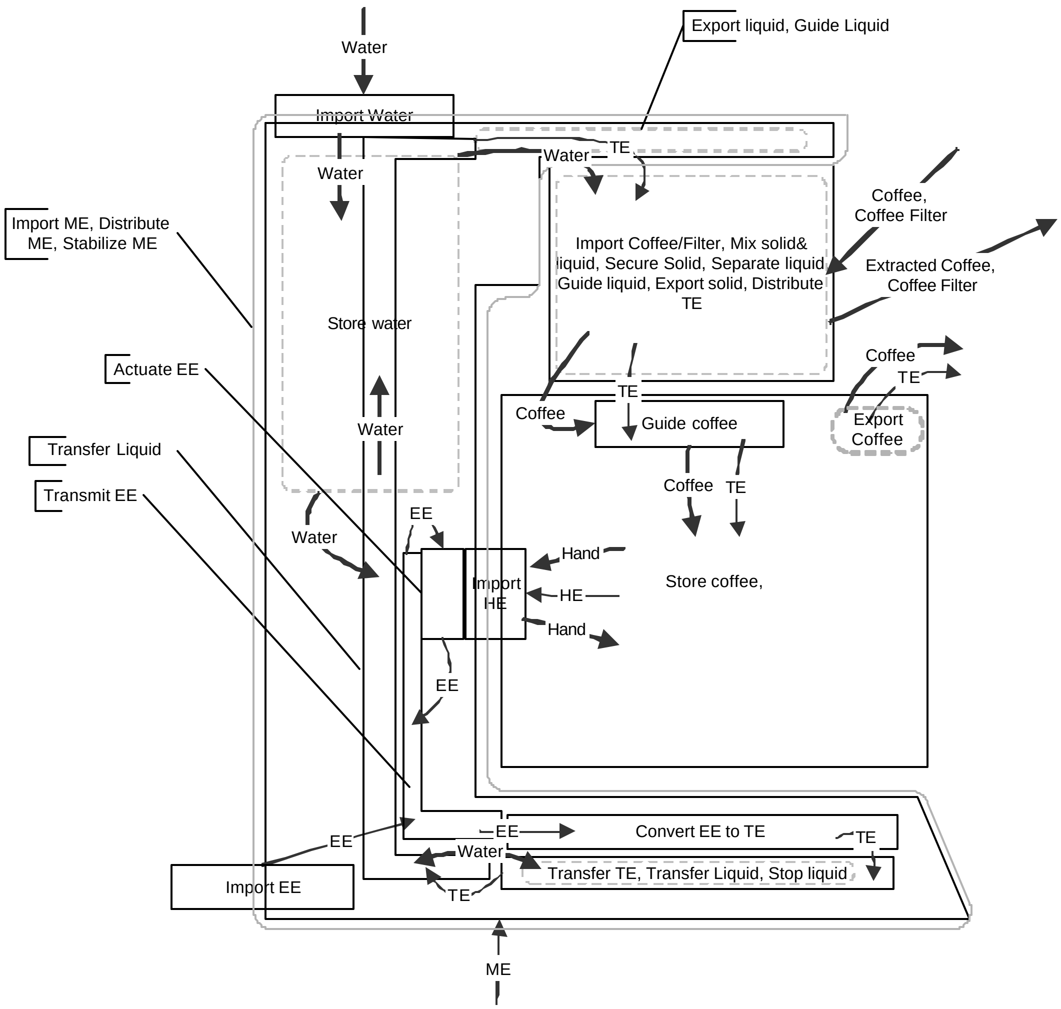


Figure C.28 Mr. Coffee maker

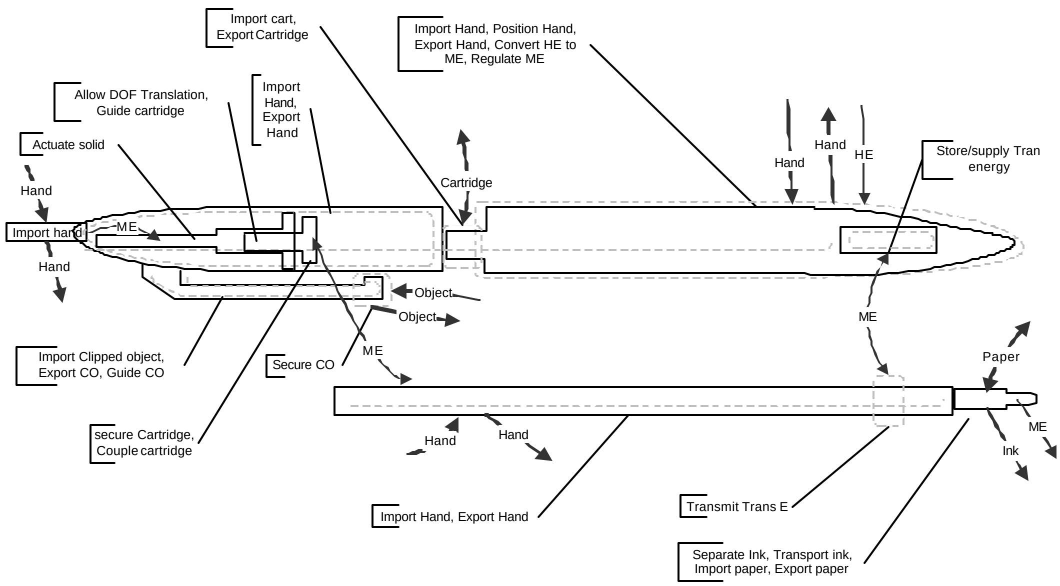


Figure C.29 Bic ball pen

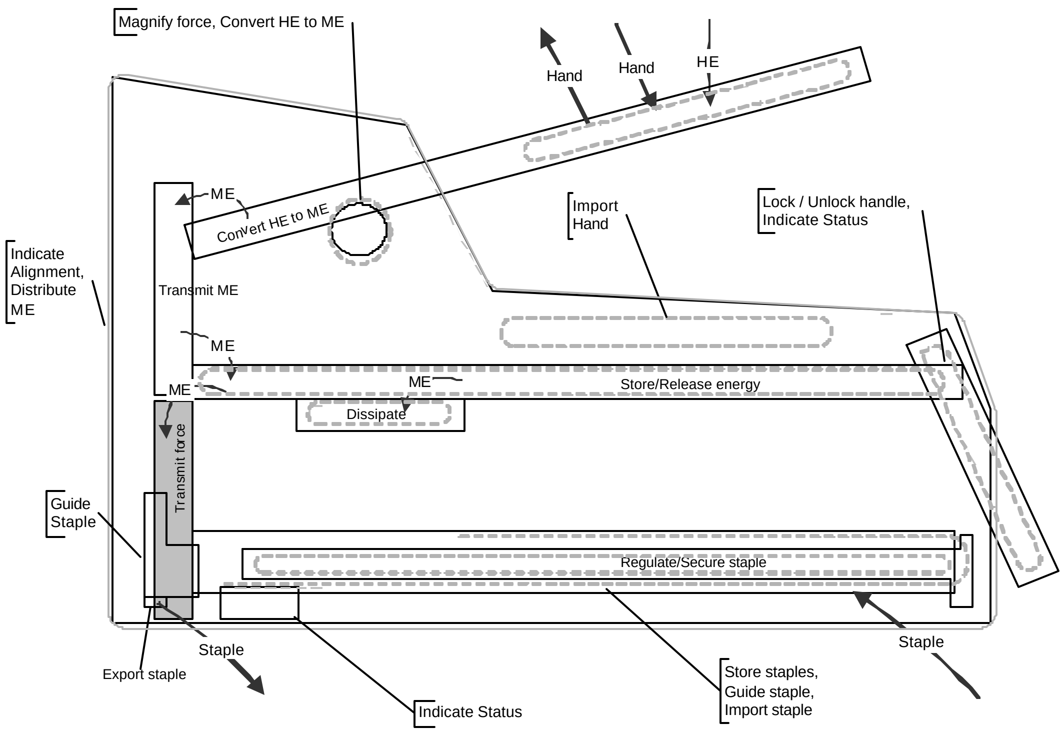


Figure C.30 Arrow light duty stapler

Appendix D: Branch Diagrams for 30 Products

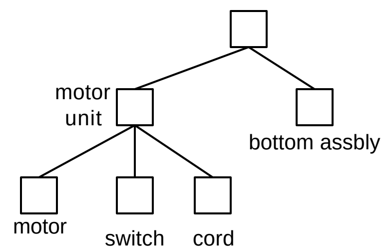


Figure D.1 GE hand blender

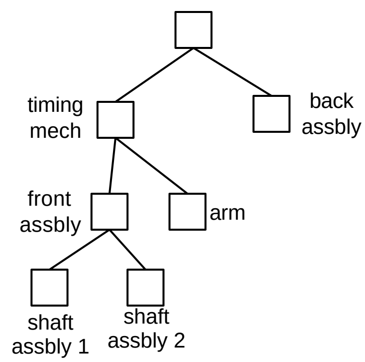


Figure D.2 Metronome

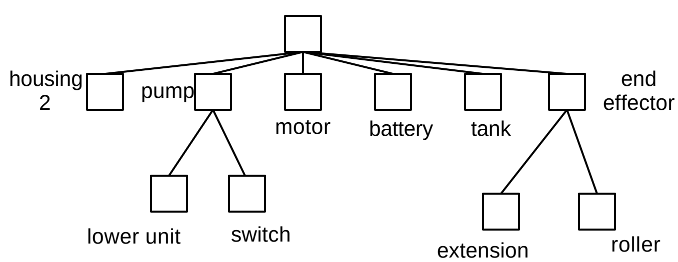


Figure D.3 Wagner paint roller

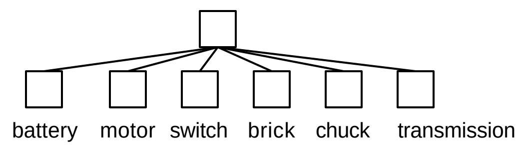


Figure D.4 Skil Twist screwdriver

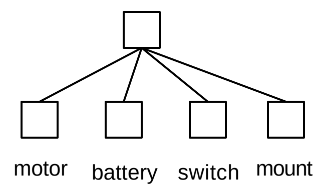


Figure D.5 Freedom cordless hand sander

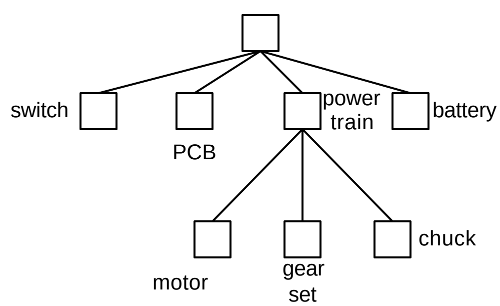


Figure D.6 Handi-Works mini drill

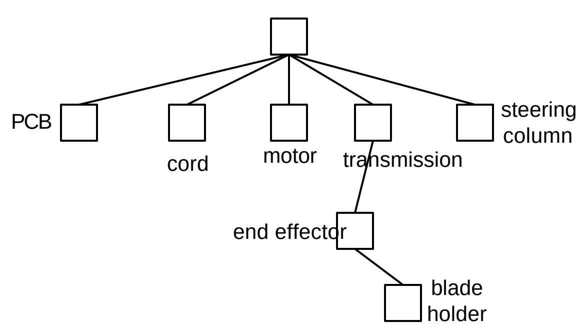


Figure D.7 Black and Decker Jigsaw

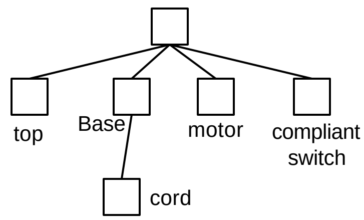


Figure D.8 Braun coffee grinder

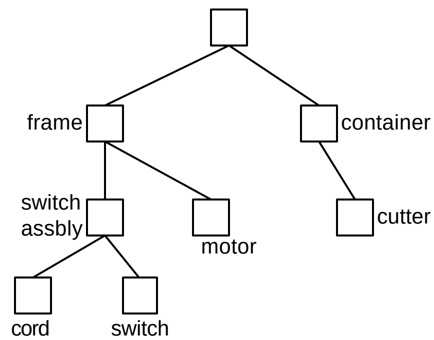


Figure D.9 Black and Decker Handy Chopper

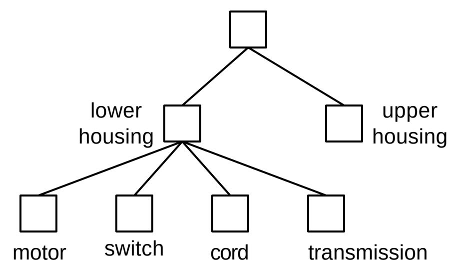


Figure D.10 Toastmaster electric knife

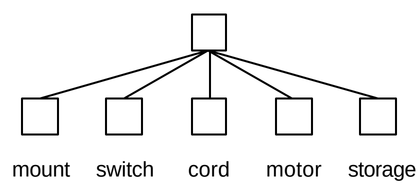


Figure D.11 DeWalt palm sander

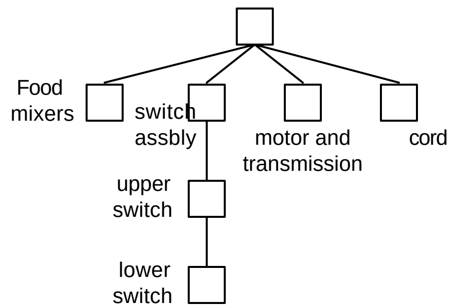


Figure D.12 Black and Decker hand mixer

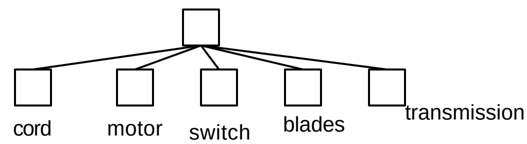


Figure D.13 Black and Decker electric knife

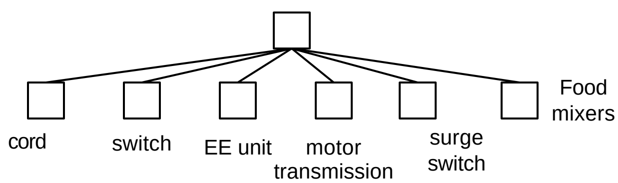


Figure D.14 GE hand mixer

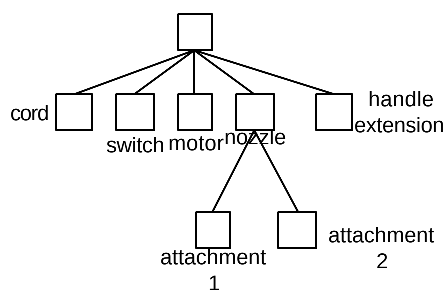


Figure D.15 Bissel hand vacuum

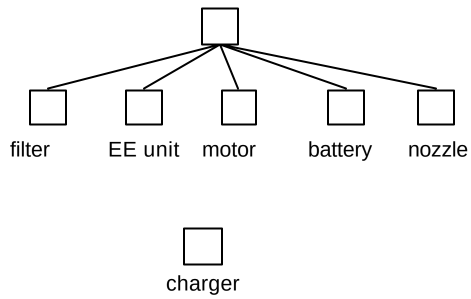


Figure D.16   Dustbuster

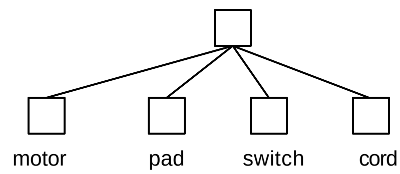


Figure D.17   Black and Decker hand sander

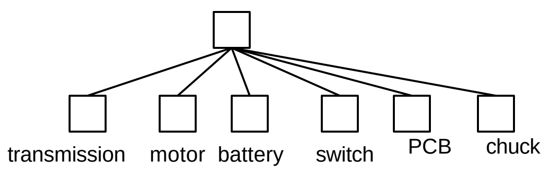


Figure D.18   Handi-Works screwdriver

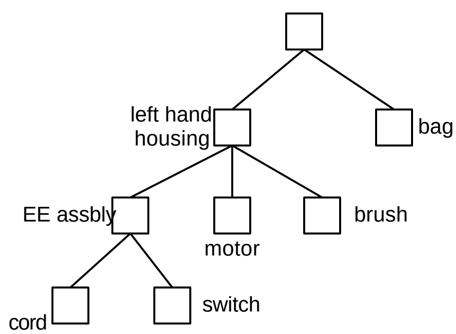


Figure D.19   Dirt Devil Spot Scrubber

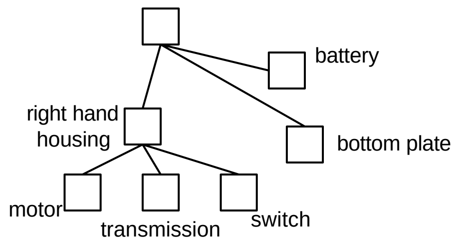


Figure D.20   Versa-Pak saber saw

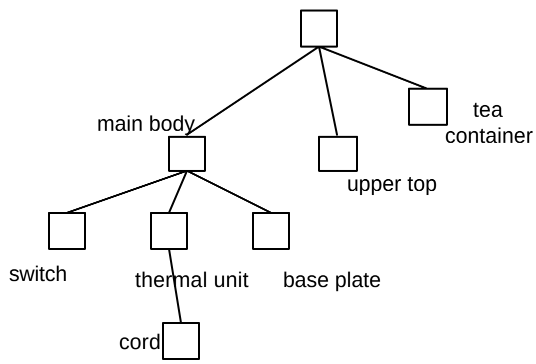


Figure D.21   Mr. Coffee ice tea maker

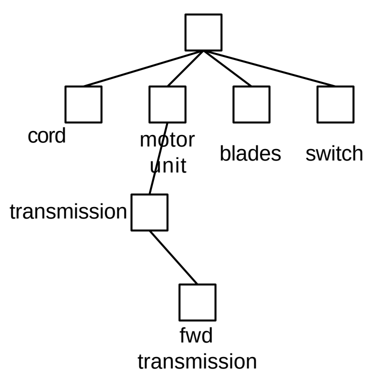


Figure D.22   GE electric knife

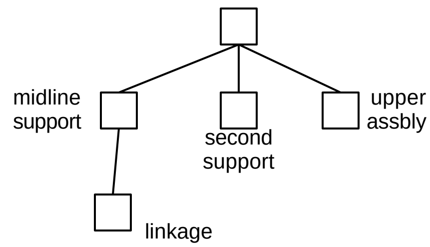


Figure D.23 Metro weighing scale

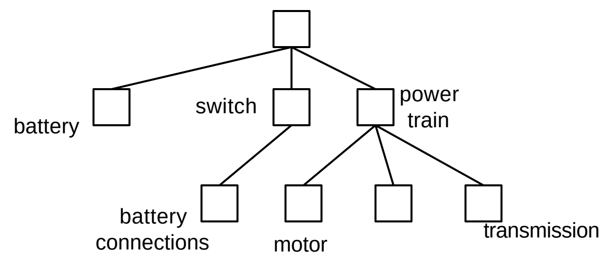


Figure D.24 Black and Decker cordless drill

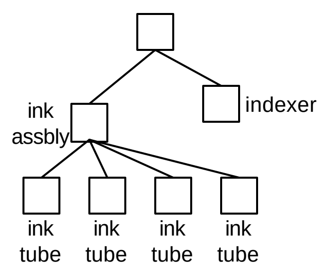


Figure D.25 3-in-one pen

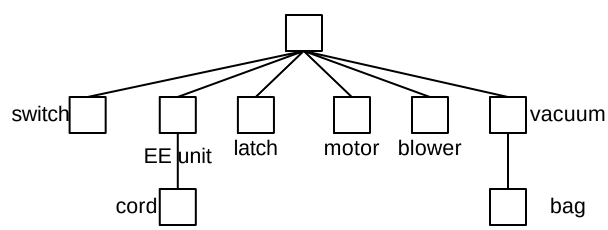


Figure D.26 Black and Decker leaf blower

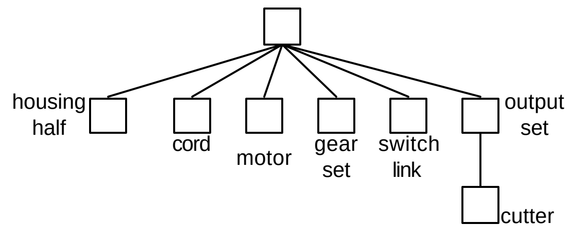


Figure D.27   Presto salad shooter

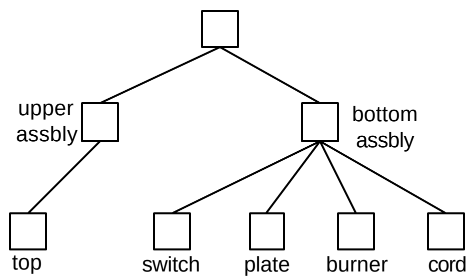


Figure D.28   Mr. Coffee maker

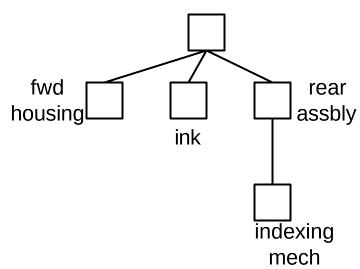


Figure D.29   Bic ball pen

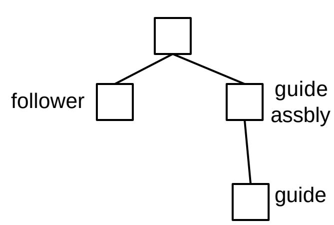


Figure D.30   Arrow light duty stapler

Bibliography

Aguirre-Esponda, G. 1992. *Evaluation of Technical Systems at the Design Stage*. PhD Thesis, Cambridge: Cambridge University.

Allen, K. R., Carlson-Skalak, 1998. "Defining Product Architecture During Conceptual Design." *ASME Design Engineering Technical Conference Proceedings, DETC98/DTM-5650.*

Andreasen, M. M., Kahler, S., Lund, T. 1988. *Design for Assembly*. 2nd ed. United Kingdom: ISF Publications.

Antonsson, E., Cagan, J. 2001. *Formal Engineering Design Synthesis*. United Kingdom: Cambridge University Press.

Ashby, M. F. 2000. *Materials Selection in Mechanical Design*. 2nd ed. Oxford: Butterworth Heinemann.

Blackenfelt, M. 2001. *Managing Complexity by Product Modularisation: Balancing the Aspects of Technology and Business During the Design Process*. PhD Thesis, Stockholm: Royal Institute of Technology.

Blessing, L., Chakrabarti, A., Wallace, K. 1998. "An Overview of Descriptive Studies in Relation to a General Design Research Methodology." *Designers – The Key to Successful Product Development*. London, UK: Springer.

Boothroyd, D., Dewhurst P., Knight, W. 1994. *Product Design for Manufacturing and Assembly*. New York: Marcel Dekker, Inc.

Boothroyd, G. and P. Dewhurst, 1984. "Design for Assembly: Manual Assembly," *Machine Design.*Cleveland, OH: Penton Publishing Inc.

Boston, O. P., Court, A. W., Culley, S. J., McMahon, C. A. 1998. "Design Information Issues in New Product Development." *The Design Productivity Debate.* London: Springer.

Browne, J. 1984. “Classification of Flexible Manufacturing Systems.” *The FMS Magazine: flexible manufacturing systems*. Kempston, Bedford: IFS Publications Ltd.

Campbell, J. L., Carney, C., Kantowitz, B. H. 1998. *Human Factors Design Guidelines for Advanced Traveler Information Systems (ATIS) and Commercial Vehicle Operations (CVO)*. Virginia: National Technical Information Service.

Campbell, M. I. 2000. *The A-Design Invention Machine: A Means of Automating and Investigating Conceptual Design*. PhD Thesis, Pittsburg, PA: Carnegie Mellon University.

Campbell, M. I. and Limaye, A. 2002. “New Advances in the Functional Modeling of Electro-Mechanical Components.” *ASME Design Engineering Technical Conference Proceedings, DETC02/DTM-34007*.

Chakratarti, A. and Bligh, T. 1994. “An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part I: Introduction and Knowledge Representation.” *Research in Engineering Design*, Vol 6. London: Springer.

Cointe, C. and Matta, N. 1998. “Architecture to Handle Concurrent Engineering.” *The Design Productivity Debate*. London, UK: Springer.

Cohen, P. R. 1995. *Empirical Methods for Artificial Intelligence*. Cambridge, Massachusetts: MIT Press.

Coulter, S. L., McIntosh, M. W., Bras, B., Rosen, D., 1998. “Identification of Limiting Factors for Improving Design Modularity.” *ASME Design Engineering Technical Conference Proceedings, DETC98/DTM-5659*.

Cross, N., Christiaans, H, Dorst, K. 1996. *Analysing Design Activity*. Chichester: John Wiley and Sons.

Cutherell, D., 1996. “Product Architecture.” *The PDMA Handbook of New Product Development*. New York: John Wiley & Sons.

Dahmus, J., Gonzalez-Zugasti, J., Otto, K. 2000. “Modular Product Architecture.” *ASME Design Engineering Technical Conference Proceedings, DETC00/DTM-14565*.

Dieter, G. E. 1991. Engineering Design – *A Materials and Processing Approach*. 2nd ed. New York: McGraw-Hill.

Dunn, G. and Everitt, B.S. 1982. *An Introduction to Mathematical Taxonomy.* Cambridge: Cambridge University Press.

Erens, F. and Verhulst, K. 1997. "Architectures for Product Families." *Computers in Industry*. 33, 165-178, Elsevier.

Fellegi, I., Jarque C., Wallman, K., (Eds.), 1997, *North American Industry Classification System*. United States Office of Management and Budget.

Finke, R., Ward, T., Smith, S. 1992. *Creative Cognition – Theory, Research, and Applications*. Cambridge, Massachusetts: MIT Press.

Frankenberger, E., Badke-Schaub, P., Birkhofer (Eds) 1997. *Designers – The Key to Successful Product Development*. London: Springer.

French, M. 1994. *Invention and Evolution – Design in Nature and Engineering*. 2nd ed. Great Britain: Cambridge University Press.

French, M. 1999. *Conceptual Design for Engineers*. 3rd ed. London: Springer.

Fricke, G. 1996. "Successful Individual Approaches in Engineering Design." *Research in Engineering Design*, Vol 8. London: Springer.

Gentner, D. and Stevens, A. 1983. *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gonzalez-Zugasti, J., Otto, K., Baker, J. 2000. "A Method for Architecting Product Platforms." *Research in Engineering Design*, Vol. 12. London: Springer.

Greeno, J., 1983. "Conceptual Entities." *Mental Models*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Greer, J. 2002. *Effort Flow Analysis: A Methodology for Directed Product Evolution Using Rigid Body and Compliant Mechanisms.* PhD Thesis, Austin, TX: The University of Texas at Austin.

Groner, R., Groner, M., and Bischof, W. F., 1983. *Methods of Heuristics*. New Jersey: Lawrence Erlbaum Associates.

Hales, C. 1993. *Managing Engineering Design*. Great Britain: Longman Scientific & Technical.

Harada, M., Witkin, A., Baraff, D. 1995. “Interactive Physically-Based Manipulation of Discrete/Continuous Models.” *SIGGRAPH*’95.

Harrisberger, L. 1967. *Engineersmanship – A Philosophy of Design*. California: Wadsworth Publishing Company, Inc.

Hayes, J. R. 1989. “Cognitive Processes in Creativity.” *Handbook of Creativity, Assessment, Research, and Theory*.” New York: Plenum Publishing Corporation.

He, D. and Kusiak, A. 1996. “Performance Analysis of Modular Products.” *International Journal of Production Research*. Vol 32 (1).

Hicks, R. 1984. *A History of the 35mm Still Camera*. London. Focal Press.

Hirtz, J., Stone, R., McAdams, D., Szykman, S., Wood K. 2002. “A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts.” *Research in Engineering Design*. 13: 65-82.

http://www.toaster.org/museumintro.html - “The Toaster Museum”

Hubka, V. 1982. *Principles of Engineering Design*. London: Butterworth Scientific.

Hubka, V. Eder, E. 1996. *Design Science – Introduction to the Needs, Scope, and Organization of Engineering Design Knowledge*. London: Springer.

Hurst, K. 1999. *Engineering Design Principles*. Great Britain: Arnold.

IEEE Computer Society. 1998. “IEEE Standard for Application and Management of the Systems Engineering Process.” *IEEE 1220-1998*. New York: IEEE.

Jiao, J. and Tseng, M. 1999. “A Methodology of Developing Product Family Architecture for Mass Customization.” *Journal of Intelligent Manufacturing*. Vol. 10.

Jensen, T. 2000. "Function Integration Explained by Allocation and Activation of Wirk Elements." *ASME Design Engineering Technical Conference Proceedings, DETC00/DTM-14551.*

Jones, C. 1970. *Design Methods – Seeds of Human Futures*. London, Wiley – Interscience.

Kleespies, H. 2002. Seminar on the Joint Strike Fighter Program. University of Texas at Austin.

Koen, B. V. 1971. "The Teaching of the Methodology of Engineering to Large Groups of Non-Engineering Students." *Gulf-Southwest Section, American Society for Engineering Education*. Ruston, Louisiana: March 26, 1971.

Koen, B. V. 2003. *Discussion of the Method for Rightly Conducting the Engineer's Approach to Problem Solving*. Oxford University Press.

Kremer, R., 1998. "Visual Languages for Knowledge Representation." *KAW'98: Eleventh Workshop on Knowledge Acquisition, Modeling, and Management*, Banff, Alberta, Canada.

Kurfman, M., Stone, R., Van Wie, M., Wood, K. & Otto, K. 2000. "Theoretical Underpinnings of Functional Modeling: Preliminary Experimental Studies." *ASME Design Engineering Technical Conference Proceedings, DETC/DTM-14563*.

Lam, S. 1994. "A Theory of Interfaces and Modules I – Composition Theorem." *IEEE Transactions on Software Engineering.* Vol. 20 (1).

Liu, Y., Chakrabarti, A, Bligh, T. 1999. "Transforming Functional Solutions into Physical Solutions." *ASME Design Engineering Technical Conference Proceedings, DETC99/DTM-8768.*

Markman, A. B. 1999. *Knowledge Representation*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Mish, F. C.(Ed.) 1989. *Webster's Ninth New Collegiate Dictionary*. Springfield, MA: Merriam-Webster Inc.

Mitchell, W. J. 2001. "Vitruvius Redux: Formalized Design Synthesis in Architecture." *Formal Engineering Design Synthesis*. Cambridge: Cambridge University Press.

Newcomb, T. and Spurr T. 1989. *A Technical History of the Motor Car*. Bristol, England: Adam Hilger.

Norman, D. 1983. "Some Observations on Mental Models" *Mental Models*. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Nowack, M. 1997. *Design Guideline Support for Manufacturability*. PhD Thesis, Cambridge: Cambridge University Press.

Ohno-Machando, L., Gennari, J., Murphy, S., Jain, N., Tu, S., Oliver, D., Pattison-Gordon, E., Greenes, R., Shortliffe, E., Barnett, G., 1998. "The Guideline Interchange Format: A Model for Representing Guidelines." *Journal of the American Medical Informatics Association* 5(4):357-372.

Otto, K. and Wood, K. 2001. *Product Design – Techniques in Reverse Engineering and New Product Development*. New Jersey: Prentice Hall.

Pahl, G. and Beitz, W. 1996. *Engineering Design: A Systematic Approach*. New York: Springer-Verlag.

Palani Rajan, P. K., Van Wie, M., Otto, K., Wood, K. 2003. "Designing for Flexibility – Measures and Guidelines." *Proceedings from International Conference on Engineering Design*, Stockholm, Sweden.

Pedersen, K., Emblemsvag, J. Baily, R., Allen, J., Mistree, F. 2000. "Validating Design Methods and Research: The Validation Square." *ASME Design Engineering Technical Conference Proceedings, DETC00/DTM-14579*.

Pugh, S. 1991. *Total Design – Integrated Methods for Successful Product Engineering*. England: Addison-Wesley Publishing.

Pugh, S. 1996. *Creating Innovative Products Using Total Design*. Massachusetts: Addison-Wesley Publishing Company.

Reddy, J. M., Finger, S., Konda, S., Subrahmanian, E. 1998. "Design as Building and Reusing Artifact Theories: Understanding and Supporting Growth of Design Knowledge." *The Design Productivity Debate*. London: Springer.

Reich, Y. 2002. “Synthesis and Theory of Knowledge: General Design Theory as a Theory of Knowledge, and its Implication to Design.” *Engineering Design Synthesis: Understanding, Approaches, and Tools*. London: Springer.

Roozenburg, N. F. M., Eekels, J. 1995. *Product Design: Fundamentals and Methods*. Great Britain: John Wiley & Sons.

Rosen, D. W., 1996 “Design of Modular Product Architectures in Discrete Design Spaces Subject to Life Cycle Issues.” *ASME Design Engineering Technical Conference Proceedings, DETC96/DAC-1485.*

Russell, W. and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall.

Sander, I. and Jantsch, A., 1999. “Formal System Design Based on the Synchrony Hypothesis, Functional Models, and Skeletons.” *IEEE 12th International Conference on VSLI Design*.

Shenhar, A. 1998. “From Theory to Practice: Toward a Typology of Project-Management Styles.” *IEEE Transactions on Engineering Management*. Vol 45. 33-48.

Slocum, A. 1992. *Precision Machine Design*. New Jersey: Prentice Hall.

Sosa, M., Eppinger, S., Rowles, C., 2000. “Designing Modular and Integrative Systems.” *ASME Design Engineering Technical Conference Proceedings, DETC00/DTM-14571*.

Stephenson, J. 1995. *Design for Reliability in Mechanical Systems*. PhD Thesis, Cambridge: Cambridge University.

Sternberg, R. 1999. “What Do We Know About Tacit Knowledge? Making the Tacit Become Explicit.” *Tacit Knowledge In Professional Practice*. Mahwah, NJ: Lawrence Erlbaum Associates.

Stiny, G. 2001. “How to Calculate with Shapes.” *Formal Engineering Design Synthesis*. Cambridge: Cambridge University Press.

Stone, R., 1997. *Towards a Theory of Modular Design*. PhD Thesis, Austin, TX: The University of Texas at Austin.

Stone, R., Wood, K., Crawford, R., 1998. "A Heuristic Method to Identify Modules from a Functional Description of a Product." *ASME Design Engineering Technical Conference Proceedings, DETC98/DTM-5642.*

Suh, N. 1990. *The Principles of Design*. New York: Oxford University Press.

Thompson, D. 2000. *Feasibility of a Skeletal Modeler for Conceptual Mechanical Design*. PhD Thesis, Austin, TX: The University of Texas at Austin.

Tomiyama, T., Yoshioka, M., and Tsumaya, A., 2002. "A Knowledge Operation Model of Synthesis." *Engineering Design Synthesis: Understanding, Approaches, and Tools*. London: Springer.

Ullman, D. 1997. *The Mechanical Design Process*. 2nd ed. New York: McGraw-Hill.

Ulrich, K. and Eppinger, S. 2000. *Product Design and Development*. Boston: Irwin McGraw-Hill.

Van Wie, M., Greer, J., Campbell, M., Stone, R., Wood, K. 2001. "Interfaces and Product Architecture." *ASME Design Engineering Technical Conference Proceedings, DETC01/DTM-21689.*

Waldron, M. B., and Waldron, K. J. 1996. "Methods of Studying Mechanical Design." *Mechanical Design Theory and Methodology*. New York: Springer.

Waldron, M. B., and Waldron, K. J. 1996. "Design Characterizations." *Mechanical Design Theory and Methodology*. New York: Springer.

Warell, A. V. 2001. "Design Syntactics – A Contribution Towards a Theoretical Framework for Form Design." *International Conference On Engineering Design*, ICED01 Glasgow.

Welch, R. and Dixon, J. 1992. "Representing Function, Behavior, and Structure During Conceptual Design." *ASME Design Engineering Technical Conference Proceedings, Design Theory and Methodology*.

Williams, M. D., Hollan, J. D., Stevens, A. L. 1983. "Human Reasoning about a Simple Physical System." *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Associates.

White, B. Y. and Frederiksen, J. R. 1990. "Causal Model Progressions as a Foundation for Intelligent Learning Environments." *Artificial Intelligence*, 42(1), 99-157.

Wood, K. L., and Greer, J. L. 2002. "Function-Based Synthesis Methods in Engineering Design." *Formal Engineering Design Synthesis*. Cambridge, United Kingdom: Cambridge University Press.

Yoshikawa, H. 1993. "Systematization of Design Knowledge." *Annals of CIRP*. 42 (1), 131-134.

Yu, J., Gonzales-Zugasti, J., Otto, K. 1998. "Product Architecture Definition Based Upon Customer Demands." *ASME Design Engineering Technical Conference Proceedings, DETC98/DTM-5679.*

Zamirowski, E. and Otto, K. 1999. "Identifying Product Portfolio Architecture Modularity Using Function and Variety Heuristics." *ASME Design Engineering Technical Conference Proceedings, DETC99/DTM-8760.*

Zusman, A. and Zlotin, B. 1999. "Overview of Creative Methods." *The TRIZ Journal*. (online only - http://www.triz-journal.com/) July, 1999.

Zwicky, F. 1948. "The Morphological Method of Analysis and Construction." *Courant*, Anniversary Volume.

Vita

Michael James Van Wie was born on October 9, 1971 to Charlene and James Van Wie in Houston, Texas. After graduating Katy High School, he attended Texas A&M in 1990. Before graduating with a Bachelor of Science degree in mechanical engineering in 1994, he worked for 8 months at JPL in Pasadena as a coop student in 1993. He entered the University of Texas at Austin in 1995 and earned a Master of Science in Engineering degree in mechanical engineering in 1996. After working as an engineer at Eastman Chemical in Longview, TX for 2.5 years, he entered the mechanical engineering doctoral program at UT in Austin.

Permanent address: 710 E. Dean Keeton #108, Austin, TX

This dissertation was typed by the author.